

# Spatial temporal modelling of job counts data

Fitting copula-based negative binomial regression for job counts

*Leanne Dong*

*03/08/2019*

## Contents

<b>Executive Summary</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Data</b>	<b>2</b>
<b>Background</b>	<b>2</b>
Generalised Linear Model for count time series . . . . .	2
Dependency modelling: from correlation to copulas . . . . .	2
Linear regression . . . . .	3
Correlation measure . . . . .	3
Copulas . . . . .	4
Examples of Copulas . . . . .	5
<b>Gaussian Copula Marginal Regression</b>	<b>5</b>
The <code>gcmr</code> package in R . . . . .	6
Likelihood inference . . . . .	7
The <code>gcmr</code> package in R . . . . .	7
<b>Methodology</b>	<b>8</b>
Model Selection Criterion . . . . .	8
<b>Temporal dependency regression analysis</b>	<b>8</b>
<b>Conclusion</b>	<b>11</b>
Summary . . . . .	11
Limitation . . . . .	11
Future work . . . . .	11
<b>Appendix: Temporal dependency regression analysis of job ads in Brisbane</b>	<b>12</b>
<b>Reference</b>	<b>45</b>

## Executive Summary

The ability to capture data about changing demand and supply of skills in a timely and efficient manner can provide much-needed information to inform educational and employment decisions, but we need to establish whether job advertisements accurately reflect sector-specific and geographic differences in demand for skills. This project involves conducting a survey of employers which we can use to test the validity of the data and **taxonomy generated from our analysis of the job advertisements**.

In this report, I initiate the temporal modelling of stationary job count time series. The purpose of this report is to identify the best temporal regression model under the Gaussian Copula Marginal Regression

framework. Our correlation analysis demonstrates that there is a strong presence of temporal dependency between consecutive time point. We concentrate our analysis on two occupations: Professionals and Managers. Based on our trend analysis, we found that, for managers, there are consistent temporal change points across ACT, Melbourne, Perth and Sydney where the increasing trend starts to flatten and become decreasing. For professionals, there are consistent temporal change points across ACT, Adelaide, Melbourne, Perth and Sydney where the increasing trend starts to flatten and become decreasing. The dependency of discrete time series  $X_t$  and  $X_{t-1}$  is conveniently modelled in terms of multivariate normal errors. In the five regional reports (all in html format), we demonstrate how fitting of data could be improved when the dependence between consecutive counts is described through a copula function. Specifically, Temporal dependence of job counts is modelled through the error process by assuming an autoregressive moving average (ARMA) model of order  $p$  and  $q$ .

## Introduction

The purpose of this report is to model job counts from different geographical locations and occupation groups in Australia. Our data sets represent time series of monthly job counts across different cities/geographical locations of Australia during 2015-2019.

To model monthly job counts for a particular occupation group, we perform regression analysis. In particular, we use the Gaussian copula negative binomial regression, which not only control the overdispersion in our data, but also account for the dependence structure of the discrete time series data at different time of observation. We implement Gaussian copulas marginal regression with the R package `gcmr` (Masarotto and Varin, 2012) which is a framework that is not only capable to model count time series, but also provides a convenience structure to deal with different types of dependence in the regression models.

## Data

In our temporal analysis we consider monthly job counts over 8 occupation groups (under ANZSCO1digit\_BERT) for five major cities of Australia during 2015-2019. In our spatial analysis, we consider monthly job counts over 88 regions (under the SA4\_ABS\_MATCH location classifier).

## Background

### Generalised Linear Model for count time series

Here we mainly concern with the zero-inflated negative binomial regression (ZINB). ZINB is used for count data that exhibit overdispersion and excess **zeros**. For the background of GLM for count time series, we refer readers to the PhD thesis of Liboschik (2016).

### Dependency modelling: from correlation to copulas

**Question** : What are the available tool to model dependence structure?

The most common way to measure dependency is linear regression, with the use of correlation coefficients.

## Linear regression

Suppose we have  $d$  Gaussian random variables. For brevity let  $d = 2$ , then to couple them together we simply multiply to obtain the joint density function

$$f_{X_1, X_2}(x_1, x_2) = f_{X_1}(x_1) \cdot f_{X_2}(x_2)$$

Suppose our random variables are dependent but both Gaussian, to join the marginals we could use a bivariate normal

$$f(x_1, x_2, \rho) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{z}{2(1-\rho^2)}\right)$$

where

$$z \approx \frac{(x_1 - \mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1 - \mu_1)(x_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2}$$

and

$$\rho \equiv \text{corr}(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\sigma_1\sigma_2}}$$

is the correlation between  $x_1$  and  $x_2$  and  $\text{Cov}(X_1, X_2)$  is the covariance.

## Correlation measure

There are three measures for correlations.

### 1. Pearson's correlation measure

- This is a standard measure that captures degree of linear relationship. Mathematically, this is defined by

$$\rho(X_i, X_j) = \rho_{ij} = \frac{\text{Cov}(X_i, X_j)}{\sqrt{\text{Var}(X_i)\text{Var}(X_j)}}$$

- Value between  $-1$  and  $1$ .
- This is a measure that captures degree of linear relationship.
- It does not reveal all the information on the dependence structure of random couples.

There are many more short-coming of the Pearson correlation. Let us name some of them below. ] ‘\* The coefficient looks for a linear relationship. It can be fallacious when two variables have a nonlinear relationship.

- The correlation analysis assumes that all observations are independent with each other. There is an if our data include a repetitive observation on any individual.
- With small sample size, the correlation will falsely detect a relationship even though none exists.
- Pearson correlation analysis only works when variables are measured on a continuous scale.

In the case of non-Gaussianity, we have two other alternatives correlation measures.

### 2. Spearman rank correlation

This is used as a replacement of Pearson  $\rho$  in the presence of nonlinearity on the scatterplot.

- Mathematically defined as

$$r(X_i, X_j) = r_{ij} = \rho(F_i(X_i), F_j(X_j))$$

where  $F_i$  and  $F_j$  are the respective marginal distributions.

- Value between  $-1$  and  $1$ .

- It is indeed Pearson  $\rho$  but applied to the transformed variable  $F_i(X_i)$  and  $F_j(X_j)$ 
  - It can be shown:

$$\begin{aligned} r(X_i, X_j) &= 12 \iint [F(X_i, X_j) - F_i(X_i)F_j(X_j)] dF_i(X_i) dF_j(X_j) - 1 \\ &= 12\mathbb{E}[F(X_i, X_j) - F_i(X_i)F_j(X_j)] - 1 \end{aligned}$$

3. Kendall  $\tau$ :

- Mathematical defined as

$$\tau(Z_i, Z_j) = \mathbb{P}[(X_i - X'_i)(X_j - X'_j) > 0] - \mathbb{P}[(X_i - X'_i)(X_j - X'_j) < 0]$$

where  $(X_i, X_j)$  and  $X'_i, X'_j$  are two independent realisation. The first term is called the probability of concordance; the latter, probability of discordance.

- Usually smaller values than Spearman rho correlation.
- Calculations based on concordant and discordant pairs. Insensitive to error.
- $p$  values are more accurate with smaller sample sizes.
- It can be shown:  $\tau(X_i, X_j) = 4\mathbb{E}(F(X_i, X_j)) - 1$

We now introduce the more rigorous, systematic treatment for dependency using copula theory.

## Copulas

Copulas are objects from probability theory allows modelling dependence of two or more random variables. A copula is a function that connects the marginal to joint distribution. It is used to combine marginals of individual r.v. to arrive a joint distribution of a collection of r.v.. With help of copula, one can dissociate the marginal distributions from the joint distribution and therefore, focus on only the statistical dependence between variables. The heart of Copulas theory is the Sklar's Theorem, stating that any multivariate distribution can be expressed as the copula function evaluated at each of the marginal distributions. More precisely, suppose  $X = (x_1, x_2, \dots, x_d)$  is a random vector with corresponding distribution  $F$  on  $\mathbb{R}^d$ . The copula associated with  $F$  is a distribution function  $C : [0, 1]^d \rightarrow [0, 1]$  with uniform marginals such that

$$F(x_1, x_2, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$$

Now if all  $F_i$  continuous. (By continuous we mean that if  $F$  is a continuous distribution on  $\mathbb{R}^d$  with univariate marginals  $F_1, \dots, F_d$ ), then  $C(\mathbf{u}) = F(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))$  is unique. Conversely, for given  $C$  and univariate CDF  $F_i$ ,  $F$  in above is a multivariate CDF with marginals  $F_1, \dots, F_d$ . Consequently,

$$C(\mathbf{u}) = F(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))$$

If you ask a statistician or actuary what a copula is they might tell you “a copula is a multivariate distribution  $C(U_1, U_2, \dots, U_n)$  such that marginalising gives  $U_i \sim \text{Uniform}(0, 1)$ ”.

A few remarks from the Sklar theory,

1. If we have two dependent but non Gaussian r.v., then the joint pdf would be

$$f(x_1, x_2) = f(x_1)f(x_2)c$$

where  $c$  is the copula.

2. **Independent copula:**  $C(u_1, \dots, u_n) = u_1 \dots u_n$ .
3. Other copulas generally contain parameters that describe the dependence.

4. Copula captures the dependence structure, while separating the effects of the marginal.

Copulas are primarily used in the context of actuarial pricing and risk theory. It has also been a very popular tool in financial risk management and credit portfolios. There are many excellent papers introduce copulae with examples. For more detailed the book by Nelson (2006) is highly recommended.\

### Motivation

The needs for copulae come from the needs of modelling dependence.

### Common fallacy

A small correlation  $\rho(X_1, X_2)$  implies that  $X_1$  and  $X_2$  are close to being independent

- Wrong! Independence implies zero correlation. But the converse is not always true. Mathematically,  $\rho(X_1, X_2) = 0$  implies  $X_1 \perp\!\!\!\perp X_2$ . However two **uncorrelated normally** distributed random variables can still be dependent under certain construction, as the following example stands.
- *Example 1:* Company's two risk assets  $X_1$  and  $X_2$ . Let  $Z \sim N(0, 1)$  and Let  $U$  be the economic stress generator, independent of  $Z$ . Moreover,  $\mathbb{P}(U = -1) = 1/2 = \mathbb{P}(U = 1)$ . Consider

$$X_1 = Z \sim N(0, 1)$$

and

$$X_2 = UZ \sim N(0, 1).$$

Now  $\text{Cov}(X_1, X_2) = \mathbb{E}(X_1, X_2) = \mathbb{E}(UZ^2) = \mathbb{E}(U)\mathbb{E}(Z^2) = 0$  hence  $\rho(X_1, X_2) = 0$ . But, isn't that clear  $X_1$  and  $X_2$  are strongly dependent, with 50% probability co-monotone and 50% probability counter-monotone.

### Examples of Copulas

Some common copulae are Gaussian,  $t$ , Archimedean (Clayton, Frank, Gumbel, Cook-Johnson) etc.

1. **Elliptic copulas** are copulas of elliptical distributions. Two main types are Gaussian and student  $t$  copula.

- Gaussian copula:  $C(u_1, \dots, u_d) = \Phi_{\Sigma}(\phi^{-1}(u_1), \dots, \phi^{-1}(u_d))$
- Student  $t$  copula:  $C(u_1, \dots, u_d) = T_{\Sigma, \nu}(t_{\nu}^{-1}(u_1), \dots, t_{\nu}^{-1}(u_d))$

2. **Archimedean copulas** has the form

$$C(u_1, \dots, u_d) = \psi^{-1}(\psi(u_1) + \dots + \psi(u_d))$$

for all  $0 \leq u_1, \dots, u_d \leq 1$  and for some function  $\psi$  (the so-called generator) satisfying: 1)  $\psi(1) = 0$ ; 2)  $\psi$  is decreasing; and 3)  $\psi$  is convex.

## Gaussian Copula Marginal Regression

Let  $\mathbf{X} = (X_1, \dots, X_8)^T$  be a vector of 8 correlated responses of discrete type. We consider the problem of modelling how the distribution of  $X_i$  varies accordingly to changes on a  $p$ -dimensional vector of covariates  $m_i$ . Dependence among the  $X_i$  is considered a secondary aspect but is important for obtaining correct inference conclusion.

Now we would like to specify a marginal parametric regression model. Let us denote by  $F_i(x_i; \mu) = \mathbb{P}(X_i \leq x_i | m_i; \mu)$  the cumulative distribution function of the response  $X_i$  given  $\mathbf{m}_i$  for a vector of unknown parameters  $\mu$ . In general terms, the regression model that relates  $X_i$  to  $m_i$  can be written as

$$X_i = g(\mathbf{m}_i, \epsilon_i; \mu), \quad i = 1, \dots, 8$$

where  $g(\cdot)$  is a suitable function,  $\epsilon_i$  is a stochastic variable commonly termed as the error, and  $\mu$  is a vector of unknown parameters. Among various possible specifications for  $g(\cdot)$ , a useful choice is

$$X_i = F_i^{-1}(\Phi(\epsilon_i); \mu), \quad i = 1, \dots, 8$$

where  $\epsilon_i$  is a standard normal variate and  $\Phi(\cdot)$  its cumulative distribution function. The above expression makes use of the probability integral transformation in order to assure that  $X_i$  given  $\mathbf{m}_i$  has the desired distribution  $F_i(x_i; \mu)$  and the error variable  $\epsilon_i$  is specified in the familiar terms of a normal variable.

To fit copula, we apply the **gcmr** R package develop by (Masarotto and Varin 2012), which gives a flexible framework which is also capable to model count time series. Under this framework, we model our marginal distribution of a time series  $X_t$  given a covariate vector  $m_t$  by Poisson or Negative Binomial distribution with mean  $\mu_t$  using that  $g(\mu_t) = \beta_0 + \eta^T m_t$ . Moreover, we introduce randomness through an unobserved error process  $\epsilon_t$  by assuming the inverse, i.e.  $X_t = F^{-1}(\Phi(\epsilon); \mu)$  exist, where  $F_t$  is the CDF of a Negative Binomial distribution with mean  $\mu_t$  and  $\Phi$  is the CDF of the standard normal. Hence the actual value of  $X_t$  is the  $\Phi(\epsilon_t)$ -quantile of the Negative Binomial distribution with mean  $\mu_t$ .

One very attractive feature of the Gaussian copula approach is that it allows us to express various forms of dependence structure through suitable parametrization of correlation matrix. More precisely, in the **gcmr**, the dependence not captured by the covariates is modeled by complementing the regression model described by  $F_i^{-1}(\Phi(\epsilon_i); \mu)$  with the assumption that the joint distribution of the errors is multivariate normal

$$\epsilon \sim \text{MVN}(0, \Omega),$$

where  $\epsilon = (\epsilon_1, \dots, \epsilon_8)^T$  and  $\Omega$  is a suitable correlation matrix

Regression models for serially correlated observations can be constructed by assuming that  $\Omega$  correspond to the correlation matrix of a  $ARMA(p, q)$ .

## The **gcmr** package in R

The **gcmr** (version 1.01) package in R was developed by Masarotto and Varin in 2017, and currently depends on version 3.0.0 of R. The package is used to fit Gaussian copula marginal regression models. It can be found in a statistical software R (R Core Team, 2016). The main function of the **gcmr** is **gcmr()**, which takes an argument of:

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

The arguments in the function are used for model-frame specification (Masarotto & Varin, 2017). The formula is used to restrict analysis to a subset of the data, to set an offset, or to fix contrasts for factors. The *marginal* and *cormat* specify the marginal part of the model and the copula correlation matrix. The *start* is used for supplying the starting values, while the *fixed* is used for fixing the values of some parameters in the model building and the options used to set the fitting options that affect the fitting of GCMR models. We remark that, the default optimization algorithm *optim* fails in our case. We had to write our own helper function (see *myopt*) for this purpose

```
options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100, 1000),
no.se = FALSE, opt = myopt)
```

## Likelihood inference

In this section, we review maximum likelihood inference for GCMR models. The density of  $X_i$  given  $m_i$  is denoted by  $p_i(x_i; \mu) = dF(x_i; \mu)/dx_i$  and depends on a vector of *marginal parameters*  $\mathbf{m}$ . The Gaussian copula correlation matrix is parametrized in terms of a vector of *dependence parameters*  $\tau$ . Then, the whole parameter vector of the GCMR model is  $\theta = (\mu^T, \tau^T)^T$ .

In continuous case, in term of 1-1 relationship between the response variable  $X_i$  and the error  $\epsilon_i$  yields the following likelihood for  $\theta$  as

$$\mathcal{L}(\theta; \mathbf{x}) = \mathcal{L}_{\text{ind}}(\theta; \mathbf{x})q(\epsilon; \theta),$$

where  $\mathcal{L}(\theta; \mathbf{x})$  is the independence likelihood and  $q(\epsilon, \theta)$  is the measure of evidence for dependence among the errors.

For discrete cases, the likelihoods requires a computation of a  $\mathbb{R}^8$ -valued integral with  $\nu_1$  and  $\nu_8$  being the parameters to be estimated.

$$\mathcal{L}(\theta; \mathbf{x}) = \int_{\nu_1(x_1; \mu)} \cdots \int_{\nu_8(x_8; \mu)} p(\epsilon_1, \dots, \epsilon_8; \theta) d\epsilon_1 \cdots d\epsilon_8.$$

## The gcmr package in R

The `gcmr` (version 1.01) package in R was developed by Masarotto and Varin in 2017, and currently depends on version 3.0.0 of R. The package is used to fit Gaussian copula marginal regression models. It can be found in a statistical software R (R Core Team, 2016). The main function of the `gcmr` is `gcmr()`, which takes an argument of: `gcmr(formula, data, subset, offset, marginal, cormat, start, fixed, options = gcmr.options(...), model = TRUE, ...)`. The arguments in the function are used for model-frame specification (Masarotto & Varin, 2017). The formula is used to restrict analysis to a subset of the data, to set an offset, or to fix contrasts for factors. The marginal and cormat specify the marginal part of the model and the copula correlation matrix. The start is used for supplying the starting values, while the fixed is used for fixing the values of some parameters in the model building and the options used to set the fitting options that affect the fitting of GCMR models. We remark that, the default optimization algorithm `optim` fails in our case. We had to write our own helper function (see `myopt` below) for this purpose

```
options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100, 1000), no.se = FALSE, opt = myopt)
```

```
myopt <- function (start, loglik, lower, upper)
{
  require(nloptr)
  fn.opt <- function(x) {
    if (any(x <= lower || x >= upper))
      NA
    else -sum(loglik(x))
  }
  # ans <- optim(par = start, fn = fn.opt, method = method, control = control)
  start[is.na(start)] <- 0
  ans <- nloptr::lbfgs(x0 = start, fn = fn.opt,
    lower = lower, upper = upper,
    control = list(xtol_rel = 1e-8, maxeval = 10000, check_derivatives = F),
    nl.info = F)
  if (ans$convergence)
    warning(paste("optim exits with code", ans$convergence))
}
```

```
list(estimate = ans$par, maximum = ans$value, convergence = ans$convergence)
}
```

## Methodology

To fit copula, we apply the `gcmr` R package develop by (Masarotto and Varin 2012), which gives a flexible framework which is also capable to model count time series. Under this framework, we model our marginal distribution of a time series  $X_t$  given a covariate vector  $m_t$  by Poisson or Negative Binomial distribution with mean  $\mu_t$  using that  $g(\mu_t) = \beta_0 + \eta^T m_t$ . Moreover, we introduce randomness through an unobserved error process  $\epsilon_t$  by assuming the inverse, i.e.  $X_t = F^{-1}(\Phi(\epsilon); \mu)$  exist, where  $F_t$  is the CDF of a Negative Binomial distribution with mean  $\mu_t$  and  $\Phi$  is the CDF of the standard normal. Hence the actual value of  $X_t$  is the  $\Phi(\epsilon_t)$ -quantile of the Negative Binomial distribution with mean  $\mu_t$ .

## Model Selection Criterion

Following the choice from the developers of the `gcmr` package. Our model selection criterion is based upon the Akaike Information Criterion (AIC). The AIC (see Chapter 1 of Akaike (1998)) was developed by Akaike as estimators of expected Kullback-Liebr distance between the model generating the data and a fitted candidate model. More precisely, the score is given by

$$AIC = -2\ln(l) + 2k$$

where  $k$  is the number of predictor of  $l$  is the maximized likelihood value,  $2k$  includes extra predictors and  $-2$  rewards the fit between the model and the data. The model with minimum AIC value is favorable.

## Temporal dependency regression analysis

We are interested in the trend in job counts. We will examine data sets of Canberra (`ACT.counts.csv`), Sydney (`Sydney.counts.csv`), Melbourne (`Melbourne.counts.csv`), Perth (`Perth.counts.csv`), Brisbane (`Brisbane.counts.csv`) and Adelaide (`Adelaide.counts.csv`).

Here we perform exploratory data analysis to develop an understanding of our data. Let first note that the occupation codes under ANZSCO1digit\_BERT represents

- 1 Managers
- 2 Professionals
- 3 Technicians and Trades Workers
- 4 Community and Personal Service Workers
- 5 Clerical and Administrative Workers
- 6 Sales Workers
- 7 Machinery Operators and Drivers
- 8 Labourer

We examined the above different occupation. For each occupation, We loaded the dataset, we start with a glance of the cross-correlation among occupation. We found significant dependency of consecutive observation. By inspecting the Autocovariance functions (ACF) and Partial Autocovariance functions (PACF), we gain a sense of the order of Moving Average and Autoregressive terms to specify later in `arma.cormat`. (subject to the limitation of the R package)



We want to start with the best base model. To this end, we performed autonomous model selection by enumerating all possible feature sets of independent variables and dependent variables. (See R codes in the separate html reports) The independent variables are the month number and their seasonal associates:  $\text{Mth.No}$ ,  $\cos(\text{Mth.No} * 2 * \pi/12)$ ,  $\sin(\text{Mth.No} * 2 * \pi/12)$ , and the dependent variables represent the counts w.r.t. each occupation are,  $\text{dvar1}, \text{dvar2}, \text{dvar3}, \text{dvar4}, \text{dvar5}, \text{dvar6}, \text{dvar7}, \text{dvar8}$ .

We choose the best model based on the AIC criterion for each occupation. Then we fit Gaussian Copula Marginal Regression for each. By plotting the cumulative residuals of the fitted models, we found a piecewise linear trend.

For Managers: . There is consistency in the temporal trends for ACT, Melbourne, Perth and Sydney. The trend for Sydney and Perth indicate a significant change from in month 27 (May 2017) where the increasing trend starts to flatten and become decreasing. The trend for ACT and Melbourne indicates a significant change from in month 32 (Oct 2017) where the increasing trend starts to flatten and become decreasing. The trend for Adelaide indicates a significant change from in month 42 (Oct 2018) where the increasing trend starts to flatten and become decreasing. To account for the change points, we add an additional explanatory variable in each of the fitted `gcmr` model. However the coefficient of this extra interaction is found to be statistically insignificant. Moreover, the AIC is worst. Hence we retain the base model.

Hence we arrive with the final fitted model for Sydney with the optimal AICs as

```
fit1 <- gcmr(x1 ~ Mth.No + Mth.No*as.factor(Mth.No<27), data = syd,
marginal=negbin.marg, cormat = arma.cormat(2,0),
options = list(seed = round(runif(1, 1, 1e+05))),
nrep = c(100, 1000), no.se = FALSE, opt = myopt))
fit1$convergence <- 0
summary(fit1)
```

For Sydney, as refer to the Sydney's report (see separate html file), the estimation of the dispersion parameter  $\kappa$  is 0.0275, with standard error of 0.006. These values suggest evidence of overdispersion, that is, the negative binomial marginals receive more support from the data than Poisson marginals. Moreover, the estimated linear trend coefficient suggests the presence of a negative linear trend. Furthermore, the interaction term with the month 27 cut-off is also significant, which suggest an evidence of trend. Hence, we obtained mathematically, the fitted model of the job counts  $X_t$  in month  $t$  is given by  $X_t|\mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 7.41 - 0.0139t + 0.015tI_{(t<27)}$$

For Perth, the best fitted model from R is

```
fit1 <- gcmr(x1 ~ Mth.No+ Mth.No*as.factor(Mth.No < 27), data = per,
marginal=negbin.marg, cormat = arma.cormat(1,0),
options = list(seed = round(runif(1, 1, 1e+05))),
nrep = c(100, 1000), no.se = FALSE, opt = myopt))
fit1$convergence <- 0
summary(fit1)
```

Following the same argument as earlier lines, we obtained mathematically, the fitted model of the job counts  $X_t$  in month  $t$  is given by  $X_t|\mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 7.39 - 0.014t + 0.016tI_{(t<27)}$$

For Melbourne, the best fitted model from R is

```
fit1 <- gcmr(x1 ~ Mth.No + Mth.No*as.factor(Mth.No <32), data = mel,
marginal=negbin.marg, cormat = arma.cormat(2,0),
options = list(seed = round(runif(1, 1, 1e+05))),
nrep = c(100, 1000), no.se = FALSE, opt = myopt))
```

```
fit1$convergence <- 0
summary(fit1)
```

Following the same argument as earlier lines, we obtained mathematically, the fitted model of the job counts  $X_t$  in month  $t$  is given by  $X_t|\mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 8.66 - 0.0088t + 0.0298tI_{(t < 32)}$$

Canberra,

```
fit1 <- gcmr(x1 ~ Mth.No + Mth.No*as.factor(Mth.No < 32), data = act,
marginal=negbin.marg, cormat = arma.cormat(2,0),
options = list(seed = round(runif(1, 1, 1e+05))),
nrep = c(100, 1000), no.se = FALSE, opt = myopt))
fit1$convergence <- 0
summary(fit1)
```

Following the same argument as earlier lines, we obtained mathematically, the fitted model of the job counts  $X_t$  in month  $t$  is given by  $X_t|\mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 6.97 - 0.011t + 0.029tI_{(t < 32)}$$

For Adelaide,

```
fit1 <- gcmr(x1 ~ Mth.No, data = ade,
marginal=negbin.marg, cormat = arma.cormat(1,0),
options = list(seed = round(runif(1, 1, 1e+05))),
nrep = c(100, 1000), no.se = FALSE, opt = myopt))
fit1$convergence <- 0
summary(fit1)
```

Following the same argument as earlier lines, we obtained mathematically, the fitted model of the job counts  $X_t$  in month  $t$  is given by  $X_t|\mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 7.53 - 0.03t - 0.038tI_{(t < 42)}$$

For professionals, once again, there is consistency in the temporal trend for ACT, Adelaide, Melbourne, Perth and Sydney. the change points for Sydney, Melbourne and Perth are also 27 (May 2017) where the trend changes from roughly increasing to decreasing job ads. The trends for Perth indicate a significant change point in month 23 (Jan 2017) where the trend changes from roughly increasing to decreasing job ads. The trend for ACT again indicates a significant change from in month 32 (Oct 2017) where the increasing trend starts to flatten and become decreasing. The trend for Adelaide indicates a significant change from in month 10 (Dec 2015) where the increasing trend starts to flatten and become decreasing. To account for the change points, we add an additional explanatory variable in each of the fitted `gcmr` model. For the same rationales in the managers case, we arrive with the following regression equations for professionals.

Sydney:

$$\ln(\mu_t) = 8.14 - 0.0090t + 0.022tI_{(t < 27)}$$

Perth:

$$\ln(\mu_t) = 8.165 - 0.0094t + 0.021tI_{(t < 27)}$$

Melbourne

$$\ln(\mu_t) = 9.24 - 0.0023t + 0.028tI_{(t < 32)}$$

Canberra

$$\ln(\mu_t) = 7.37 + 0.0043t + 0.025tI_{(t < 32)}$$

Adelaide

$$\ln(\mu_t) = 7.07 + 0.0034t + 0.035tI_{(t < 10)}$$

Note that, for Canberra and Adelaide, the estimated linear trend coefficient suggests the presence of a positive linear trend.

Finally, to assess the goodness of fit for above models, we check normality of the fitted models by inspecting the corresponding QQ plots and normal probability plots, as well as the autocorrelation diagrams. The diagnostics (See separate html reports) suggest that the distribution assumptions are appropriate and that no residual serial correlation is present in the data.

## Conclusion

### Summary

Our study suggests that the Adzuna data can be useful in identifying temporal variation in identifying trends and demand for workers and skills in five major cities of Australia. There is a need to extend the analysis further to examine spatial variation at occupation level.

### Limitation

As pointed out by the authors of `gcmr` R package, copulas with discrete marginals might not be unique. Note that the model accounts for serial dependence, it does not model the conditional distributions of  $X_t$  given the past but only its time-varying marginal distribution. The mean  $\mu_t$  of this marginal is not influenced by the actual value of the error  $\epsilon_t$ .

### Future work

Our final aim is to forecast the next months volume of unique job ads. This was outside the limit of what the `gcmr` package can provide us. Unlike in the case of generalised linear models, there is no `predict` method available in the `gcmr` package. To overcome this limitation, one could write own code to compute predictions. It was suggested by the authors of the `gcmr` package that one could start with computing the predictive density  $p(y_{\text{new}}|\text{observed data})$  for a range of possible values of  $y_{\text{new}}$  augmenting the data and then using the internal function `gcmr::llik`

Next, in the project we have only studied temporal variation in labor markets with incorporation of dependency of discrete time series  $X_t$  and  $X_{t-1}$  for each Occupation. In this direction we have yet to find copula parameters. Hence this becomes a immediate action plan for the future. On other hand, we would like to explore geographic variation. In particular, we could model dependency of job counts time series across different regions of Australia. One starting point is to examine the cross correlation of time series across different regions. Then we could fit dependence structure with multiple time series and estimates their copula parameters. The study of spatial modelling is an important but computational very challenging field. The most common tool is the Gaussian Random Field. However this method is not scalable for large amount of data. One innovated approach is known as the stochastic Partial differential equations (SPDE) approach. This approach is based on advanced theory of stochastic calculus, has a computational efficient Bayesian implementation in R via the package `R-INLA`. This approach requires the approximation of the solutions of SPDE, which can be achieved by the well known finite element method (FEM). For more detail, one could consult with the note Lindgren and Rue (n.d.) and the git book Krainski et al. (n.d.). (A hard copy will be available soon)

## Appendix: Temporal dependency regression analysis of job ads in Brisbane

First, where is Brisbane? It is the Sunshine coast in Australia!

First we load some necessary packages,

```
# List of packages for session
.pks<- c("tidyverse","foreign","MASS", "gcmr")
# Install CRAN packages (if not already installed)
.inst<- .pks %in% installed.packages()
if(length(.pks[!.inst])>0) install.packages(.pks[!.inst])
# load packages into session
lapply(.pks,require,character.only=TRUE)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages -----
```

```
## v ggplot2 3.2.0      v purrr  0.3.2
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   0.8.3      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0
```

```
## -- Conflicts -----
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
## Loading required package: foreign
```

```
## Loading required package: MASS
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      select
```

```
## Loading required package: gcmr
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

We need the following helper function to stay away from some optimisation issue.

```
source("scripts/nloptr.R")
```

Now load monthly job count data of 8 occupations for Brisbane.

```
bris<- read_csv("data/Brisbane.counts.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
```

```
## cols(
```

```
##   X1 = col_double(),
```

```
##   x1 = col_double(),
```

```
##   x2 = col_double(),
```

```
##   x3 = col_double(),
```

```
##   x4 = col_double(),
```

```
##   x5 = col_double(),
```

```
## x6 = col_double(),
## x7 = col_double(),
## x8 = col_double(),
## Mth.No = col_double(),
## Month = col_double()
## )
```

```
# inspect the first few row entries
head(bris)
```

```
## # A tibble: 6 x 11
##      X1      x1      x2      x3      x4      x5      x6      x7      x8 Mth.No Month
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     3    116    259    426    453    147    238    166    319     3     3
## 2     4     89    213    379    355    124    176    124    263     4     4
## 3     5    129    428    495    454    125    179    187    214     5     5
## 4     6    141    257    259    353    189    298    177    200     6     6
## 5     7    125    284    230    460    203    220    157    176     7     7
## 6     8    103    298    295    423    123    261    214    261     8     8
```

```
# inspect data structures
str(bris)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 51 obs. of  11 variables:
## $ X1      : num  3 4 5 6 7 8 9 10 11 12 ...
## $ x1      : num 116 89 129 141 125 103 128 122 103 77 ...
## $ x2      : num 259 213 428 257 284 298 344 241 313 184 ...
## $ x3      : num 426 379 495 259 230 295 362 211 250 159 ...
## $ x4      : num 453 355 454 353 460 423 420 389 422 274 ...
## $ x5      : num 147 124 125 189 203 123 182 170 123 104 ...
## $ x6      : num 238 176 179 298 220 261 223 235 203 157 ...
## $ x7      : num 166 124 187 177 157 214 229 184 179 139 ...
## $ x8      : num 319 263 214 200 176 261 162 199 182 133 ...
## $ Mth.No: num  3 4 5 6 7 8 9 10 11 12 ...
## $ Month : num  3 4 5 6 7 8 9 10 11 12 ...
## - attr(*, "spec")=
## .. cols(
## ..   X1 = col_double(),
## ..   x1 = col_double(),
## ..   x2 = col_double(),
## ..   x3 = col_double(),
## ..   x4 = col_double(),
## ..   x5 = col_double(),
## ..   x6 = col_double(),
## ..   x7 = col_double(),
## ..   x8 = col_double(),
## ..   Mth.No = col_double(),
## ..   Month = col_double()
## .. )
```

```
# dimension
dim(bris)
```

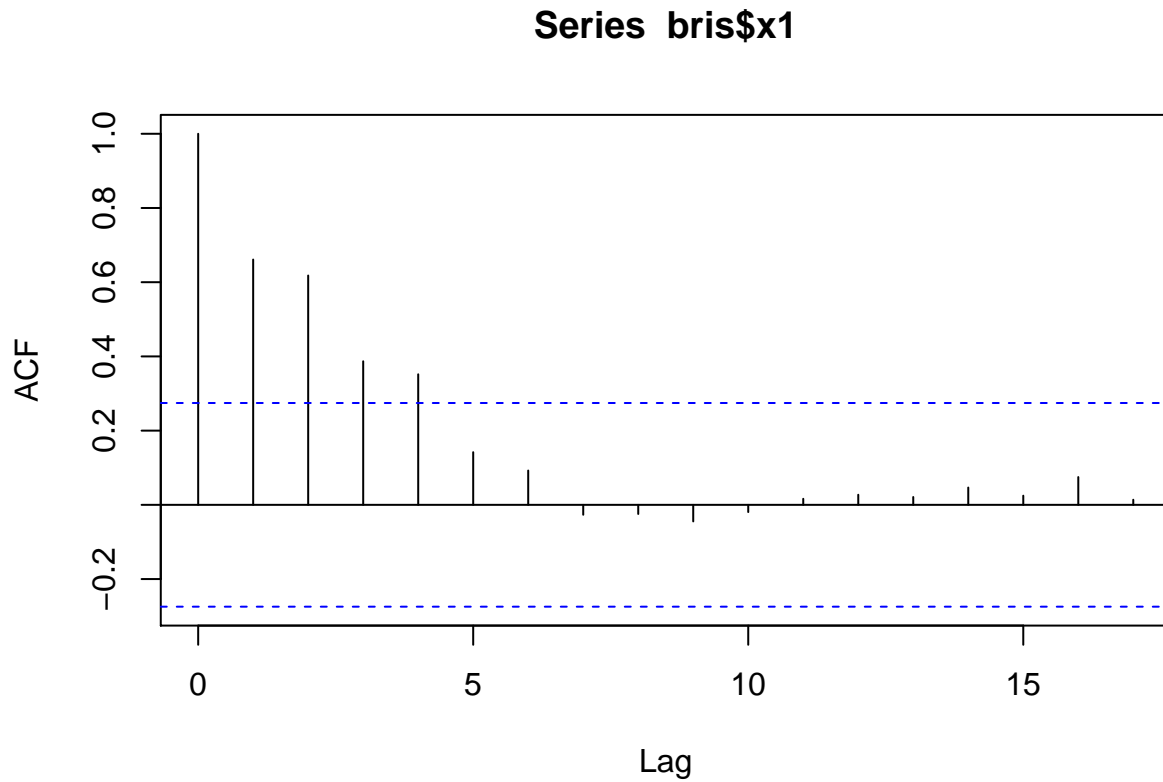
```
## [1] 51 11
```

```
# inspect variable names
names(bris)
```

```
## [1] "X1"      "x1"      "x2"      "x3"      "x4"      "x5"      "x6"
## [8] "x7"      "x8"      "Mth.No"  "Month"
```

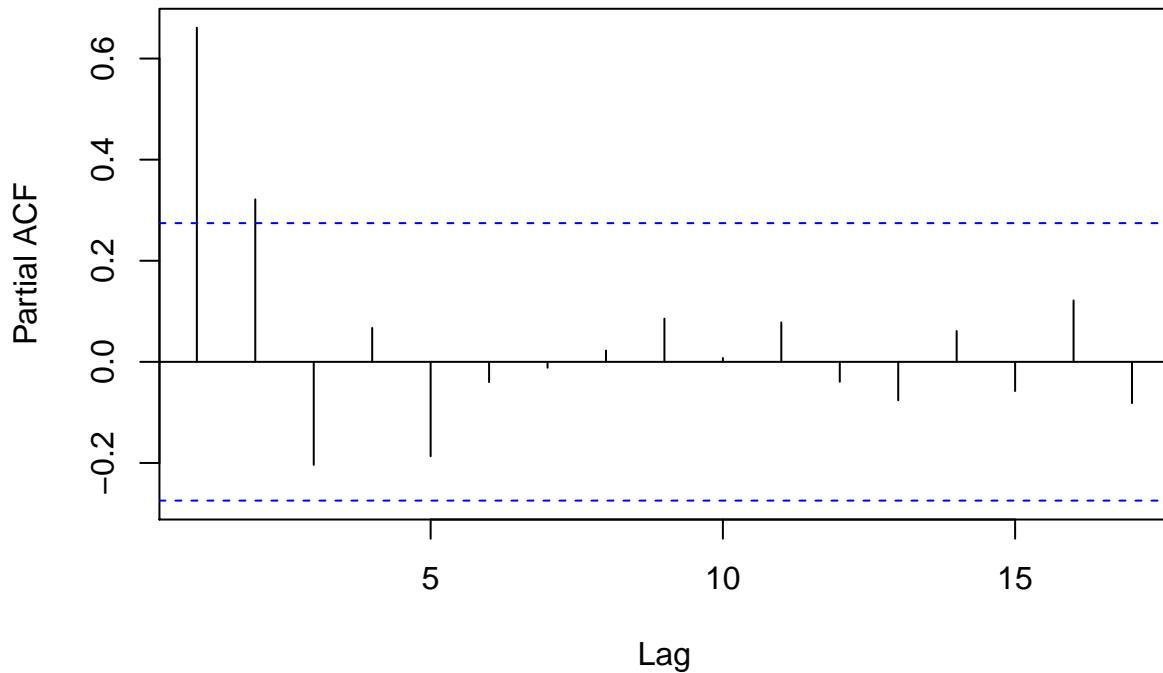
We start with inspecting the autocorrelation plots to get a sense of the correlation structure.

```
c <- acf(bris$x1,type = "correlation")
```



```
c1 <- pacf(bris$x1)
```

## Series bris\$x1



The PACF diagram suggests that ARMA(1,0)

Let us now start our fitting. This is accomplished through the method of maximum likelihood estimation using the Gaussian copula. We select the best model based on the AIC. The equation for the Gaussian copula marginal regression is as follows:

The cut off from the PACF helps us to determine the order of AR terms. The diagram suggests that ARMA(1,0).

Let us now start our fitting. This is accomplished through the method of maximum likelihood estimation using the Gaussian copula. We select the best model based on the AIC. The equation for the Gaussian copula marginal regression is as follows:

$$\mathbb{E}(X_i|m_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 m_1 + \beta_2 m_2 + \beta_3 m_3 + \epsilon_i$$

where  $p_i$  is the probability value,  $\beta_0, \beta_1, \beta_2$  are the coefficients,  $X_1, \dots, X_8$  are the predictor variables and  $\epsilon_i$  is the random error term.

Having determined the correlation structure. We ask R to run a autonomous feature enumeration to settle us the best model (based on AIC) for each Occupation.

```
dvar1 <- bris$x1
dvar2 <- bris$x2
dvar3 <- bris$x3
dvar4 <- bris$x4
dvar5 <- bris$x5
dvar6 <- bris$x6
dvar7 <- bris$x7
dvar8 <- bris$x8

ivar1 <- bris$Mth.No
```

```

temp2 <- bris %>% dplyr::select(Mth.No) %>% mutate(cosmth=cos(Mth.No * 2 * pi/12))
ivar2 <- temp2$cosmth
temp3 <- bris %>% dplyr::select(Mth.No) %>% mutate(sinmth=sin(Mth.No * 2 * pi/12))
ivar3 <- temp3$sinmth

d_vars <- paste("dvar", 1:8, sep="")
i_vars <- paste("ivar", 1:3, sep="")

# create all combinations of ind_vars
ind_vars_comb <-
  unlist( sapply( seq_len(length(i_vars)),
                  function(i) {
                    apply( combn(i_vars,i), 2, function(x) paste(x, collapse = "+"))
                  })

# pair with dep_vars:
var_comb <- expand.grid(d_vars, ind_vars_comb )

# formulas for all combinations
formula_vec <- sprintf("%s ~ %s", var_comb$Var1, var_comb$Var2)

# create models
gcmr_sum <- lapply( formula_vec, function(f) {
  fit1 <- gcmr( f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,0), options = list(seed=round(runif(1, 1, 1e+05))), nrep = c(100, 1000), no.se = FALSE, opt = myopt))
  fit1$convergence <- 0
  # fit1$coefficients <- coef( summary(fit1))
  fit1$summaries <- summary(fit1)
  return(fit1$summaries)
})

```

```
## Loading required package: nloptr
```

```
gcmr_sum
```

```

## [[1]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 4.598933   0.165032  27.867 < 2e-16 ***
## ivar1       0.014821   0.005162   2.871 0.004089 **
## dispersion  0.088675   0.025623   3.461 0.000539 ***
##
## Coefficients Gaussian copula:
##           Estimate Std. Error z value Pr(>|z|)
## ar1      0.5417      0.1212   4.468 7.89e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```



```

## log likelihood = 257.98, AIC = 523.96
##
## [[2]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.441689    0.095852  56.772 < 2e-16 ***
## ivar1        0.013338    0.002999   4.448 8.67e-06 ***
## dispersion  0.058317    0.012981   4.493 7.04e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.2695     0.1343   2.007  0.0448 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 294.73, AIC = 597.46
##
## [[3]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.878953    0.175572  33.485 < 2e-16 ***
## ivar1       -0.014213    0.005533  -2.569 0.010205 *
## dispersion  0.110019    0.029170   3.772 0.000162 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5414     0.1159   4.671 2.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 285.8, AIC = 579.59
##
## [[4]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##

```

```

## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.090683   0.061926  98.355 < 2e-16 ***
## ivar1        -0.010498   0.001968  -5.333 9.66e-08 ***
## dispersion   0.033126   0.007267   4.559 5.15e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.08191    0.13721   0.597   0.551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 282.78, AIC = 573.57
##
## [[5]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.917036   0.161949  30.362 < 2e-16 ***
## ivar1        0.005519   0.005056   1.092 0.275010
## dispersion   0.082495   0.024408   3.380 0.000725 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.5788     0.1138   5.087 3.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 257.59, AIC = 523.18
##
## [[6]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.387874   0.180323  29.879 < 2e-16 ***
## ivar1        -0.006289   0.005620  -1.119 0.26305
## dispersion   0.108550   0.031756   3.418 0.00063 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.5514     0.1201   4.591 4.4e-06 ***
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 272.17,  AIC = 552.33
##
## [[7]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.143474   0.155293  33.121 < 2e-16 ***
## ivar1        -0.004909   0.005001  -0.982  0.326
## dispersion   0.101676   0.025808   3.940 8.16e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4147     0.1300   3.189 0.00143 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 264.89,  AIC = 537.77
##
## [[8]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.562876   0.131677  42.246 < 2e-16 ***
## ivar1        -0.021583   0.004222  -5.112 3.19e-07 ***
## dispersion   0.077287   0.019120   4.042 5.29e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.3995     0.1349   2.961 0.00307 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.45,  AIC = 520.9
##
## [[9]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))

```

```

##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.02605    0.10691  47.014 < 2e-16 ***
## ivar2        -0.07966    0.09937  -0.802  0.42276
## dispersion   0.12354    0.04032   3.064  0.00218 **
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.6548     0.1051    6.23 4.67e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 260.92, AIC = 529.84
##
## [[10]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.83144    0.07517  77.582 < 2e-16 ***
## ivar2        -0.13008    0.08659  -1.502  0.133040
## dispersion   0.09161    0.02536   3.612  0.000304 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5296     0.1207    4.387 1.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 299.74, AIC = 607.49
##
## [[11]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.49583    0.11272  48.756 < 2e-16 ***
## ivar2        -0.12819    0.10958  -1.170  0.24207
## dispersion   0.14310    0.04614   3.101  0.00193 **
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)

```

```

## ar1    0.6479      0.1116    5.804 6.49e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 287.58,   AIC = 583.16
##
## [[12]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.80794    0.05432 106.929 < 2e-16 ***
## ivar2        -0.00205    0.07050  -0.029 0.976802
## dispersion   0.05705    0.01487   3.837 0.000124 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4404      0.1364   3.229 0.00124 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 290.56,   AIC = 589.12
##
## [[13]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.05783    0.07337  68.932 < 2e-16 ***
## ivar2        -0.19458    0.08224  -2.366 0.01798 *
## dispersion   0.07455    0.02262   3.296 0.00098 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5691      0.1162   4.896 9.77e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 255.44,   AIC = 518.87
##
## [[14]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,

```

```

##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.21159    0.09229  56.469 < 2e-16 ***
## ivar2        -0.06352    0.10087  -0.630  0.52886
## dispersion   0.11621    0.03391   3.427  0.00061 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5823     0.1216    4.79 1.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 272.54, AIC = 553.07
##
## [[15]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.00448    0.07513  66.608 < 2e-16 ***
## ivar2        -0.07332    0.09302  -0.788  0.430532
## dispersion   0.10648    0.02829   3.764  0.000167 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4509     0.1254    3.597 0.000322 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 265, AIC = 538
##
## [[16]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.03770    0.16506  30.521 <2e-16 ***
## ivar2        0.07222    0.11594   0.623  0.5333
## dispersion   0.20039    0.08922   2.246  0.0247 *
##

```

```

## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.7744      0.1055   7.338 2.16e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 261.96, AIC = 531.91
##
## [[17]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.03810      0.10018  50.292 < 2e-16 ***
## ivar3        -0.12205      0.09446  -1.292  0.19637
## dispersion   0.11599      0.03604   3.219  0.00129 **
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.6376      0.1068   5.968 2.4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 260.41, AIC = 528.82
##
## [[18]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.84402      0.07665  76.243 < 2e-16 ***
## ivar3        -0.05518      0.08298  -0.665  0.506094
## dispersion   0.09519      0.02531   3.761  0.000169 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5340      0.1216   4.39 1.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 300.64, AIC = 609.28
##
## [[19]]
##

```

```

## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.508177    0.119328  46.160 < 2e-16 ***
## ivar3       0.006647    0.101952   0.065  0.94802
## dispersion  0.153055    0.050387   3.038  0.00238 **
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.6677      0.1065   6.271 3.59e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 288.24, AIC = 584.48
##
## [[20]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.80282    0.05350 108.471 < 2e-16 ***
## ivar3       0.06587    0.06310   1.044 0.296489
## dispersion  0.05575    0.01463   3.812 0.000138 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4387      0.1304   3.365 0.000766 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 290.02, AIC = 588.05
##
## [[21]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08029    0.08250  61.579 < 2e-16 ***
## ivar3       -0.09588    0.08159  -1.175 0.239910

```



```

## dispersion    0.08392    0.02500    3.357 0.000789 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5927      0.1099    5.391 7.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 257.44,  AIC = 522.89
##
## [[22]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.22184    0.08449  61.803 < 2e-16 ***
## ivar3        -0.15740    0.08905  -1.768 0.077144 .
## dispersion    0.10430    0.02965    3.518 0.000435 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5523      0.1256    4.398 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 271.21,  AIC = 550.42
##
## [[23]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.01337    0.07260  69.058 < 2e-16 ***
## ivar3        -0.12169    0.08579  -1.419 0.156042
## dispersion    0.10167    0.02640    3.852 0.000117 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4360      0.1259    3.464 0.000532 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 264.35,  AIC = 536.7
##

```

```

## [[24]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.01042    0.17905  27.984 <2e-16 ***
## ivar3        0.12359    0.10852   1.139  0.2548
## dispersion   0.19339    0.07888   2.452  0.0142 *
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.7668     0.1045   7.334 2.23e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 261.48, AIC = 530.97
##
## [[25]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.592431    0.154020  29.817 < 2e-16 ***
## ivar1        0.014846    0.004821   3.080 0.002073 **
## ivar2       -0.091314    0.086933  -1.050 0.293535
## dispersion   0.082623    0.022350   3.697 0.000218 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5091     0.1295   3.931 8.46e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 257.43, AIC = 524.86
##
## [[26]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:

```

```

##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.424266   0.085506  63.437 < 2e-16 ***
## ivar1        0.013678   0.002675   5.114 3.16e-07 ***
## ivar2       -0.123433   0.055849  -2.210 0.0271 *
## dispersion   0.051539   0.011183   4.608 4.06e-06 ***
##
## Coefficients Gaussian copula:
##           Estimate Std. Error z value Pr(>|z|)
## ar1    0.2031      0.1361    1.492   0.136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 292.51, AIC = 595.03
##
## [[27]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.870666   0.165934  35.380 < 2e-16 ***
## ivar1       -0.014311   0.005234  -2.734 0.006249 **
## ivar2       -0.136746   0.092737  -1.475 0.140334
## dispersion   0.101491   0.026252   3.866 0.000111 ***
##
## Coefficients Gaussian copula:
##           Estimate Std. Error z value Pr(>|z|)
## ar1    0.5124      0.1189    4.31 1.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 284.76, AIC = 579.52
##
## [[28]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.087730   0.061182  99.502 < 2e-16 ***
## ivar1       -0.010416   0.001945  -5.355 8.56e-08 ***
## ivar2       -0.020164   0.041697  -0.484 0.629
## dispersion   0.032896   0.007208   4.564 5.02e-06 ***
##
## Coefficients Gaussian copula:
##           Estimate Std. Error z value Pr(>|z|)

```

```

## ar1 0.06834 0.13839 0.494 0.621
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 282.67, AIC = 575.34
##
## [[29]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 4.895209 0.141063 34.702 < 2e-16 ***
## ivar1 0.005716 0.004424 1.292 0.196359
## ivar2 -0.193706 0.076094 -2.546 0.010909 *
## dispersion 0.066460 0.018319 3.628 0.000286 ***
##
## Coefficients Gaussian copula:
## Estimate Std. Error z value Pr(>|z|)
## ar1 0.5177 0.1261 4.105 4.04e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 254.69, AIC = 519.38
##
## [[30]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
## 0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
## 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.384583 0.170031 31.668 < 2e-16 ***
## ivar1 -0.006313 0.005344 -1.181 0.237460
## ivar2 -0.066110 0.095355 -0.693 0.488121
## dispersion 0.105581 0.030009 3.518 0.000434 ***
##
## Coefficients Gaussian copula:
## Estimate Std. Error z value Pr(>|z|)
## ar1 0.5393 0.1237 4.361 1.29e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 271.93, AIC = 553.86
##
## [[31]]
##

```

```

## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.144822   0.147545  34.870 < 2e-16 ***
## ivar1        -0.005107   0.004742  -1.077   0.282
## ivar2        -0.077184   0.086741  -0.890   0.374
## dispersion   0.097398   0.024274   4.012 6.01e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.3872     0.1321   2.931 0.00337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 264.51, AIC = 539.02
##
## [[32]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.566936   0.133289  41.766 < 2e-16 ***
## ivar1        -0.021690   0.004266  -5.085 3.68e-07 ***
## ivar2         0.026682   0.079520   0.336   0.737
## dispersion   0.077615   0.019423   3.996 6.44e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4058     0.1369   2.966 0.00302 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.39, AIC = 522.79
##
## [[33]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)

```

```

## (Intercept)  4.623811    0.144997   31.889 < 2e-16 ***
## ivar1        0.014113    0.004529    3.116 0.001834 **
## ivar3       -0.117240    0.078426   -1.495 0.134937
## dispersion   0.078803    0.021807    3.614 0.000302 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4922     0.1261    3.902 9.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.95, AIC = 523.9
##
## [[34]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.441458    0.089773   60.614 < 2e-16 ***
## ivar1        0.013424    0.002821    4.758 1.96e-06 ***
## ivar3       -0.062515    0.057155   -1.094  0.274
## dispersion   0.056185    0.012437    4.518 6.25e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.2455     0.1364    1.801  0.0718 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 294.16, AIC = 598.31
##
## [[35]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.882204    0.177971   33.052 < 2e-16 ***
## ivar1       -0.014289    0.005586   -2.558 0.010535 *
## ivar3       -0.015380    0.089738   -0.171 0.863917
## dispersion   0.109847    0.029876    3.677 0.000236 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5409     0.1150    4.704 2.55e-06 ***

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 285.78,  AIC = 581.56
##
## [[36]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.091026   0.057946 105.116 < 2e-16 ***
## ivar1        -0.010650   0.001848  -5.764 8.21e-09 ***
## ivar3         0.062315   0.037069   1.681  0.0928 .
## dispersion   0.031089   0.006791   4.578 4.69e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.04364    0.13760   0.317   0.751
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 281.45,  AIC = 572.89
##
## [[37]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.924271   0.158490  31.070 <2e-16 ***
## ivar1         0.005447   0.004926   1.106  0.2689
## ivar3        -0.095138   0.079104  -1.203  0.2291
## dispersion   0.077430   0.021419   3.615  0.0003 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.5608     0.1186   4.727 2.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.86,  AIC = 523.72
##
## [[38]]
##
## Call:

```

```

## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.421015   0.155978  34.755 < 2e-16 ***
## ivar1        -0.007281   0.004869  -1.495 0.134795
## ivar3        -0.169248   0.081459  -2.078 0.037736 *
## dispersion    0.092190   0.024912   3.701 0.000215 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4958     0.1259   3.937 8.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 270.21, AIC = 550.41
##
## [[39]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.161178   0.140077  36.845 < 2e-16 ***
## ivar1        -0.005393   0.004481  -1.204 0.229
## ivar3        -0.125077   0.079525  -1.573 0.116
## dispersion    0.092472   0.023238   3.979 6.91e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.3677     0.1331   2.763 0.00573 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 263.74, AIC = 537.47
##
## [[40]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.55847    0.13114  42.387 < 2e-16 ***

```



```

## ivar1      -0.02165    0.00421  -5.142 2.71e-07 ***
## ivar3      0.08146    0.07571   1.076  0.282
## dispersion 0.07561    0.01898   3.984 6.76e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.4019     0.1366   2.943 0.00325 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 255.85, AIC = 521.7
##
## [[41]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.03195    0.09674  52.016 < 2e-16 ***
## ivar2       -0.08547    0.09445  -0.905 0.365505
## ivar3       -0.12564    0.09373  -1.340 0.180113
## dispersion   0.11173    0.03357   3.328 0.000876 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.6266     0.1079   5.805 6.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 260.03, AIC = 530.05
##
## [[42]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.83511    0.07398  78.872 < 2e-16 ***
## ivar2       -0.13345    0.08552  -1.560 0.11866
## ivar3       -0.06211    0.08045  -0.772 0.44012
## dispersion   0.09004    0.02385   3.775 0.00016 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.5262     0.1207   4.359 1.31e-05 ***
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 299.46,  AIC = 608.91
##
## [[43]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.495621   0.113369  48.476 < 2e-16 ***
## ivar2        -0.128095   0.111594  -1.148  0.25102
## ivar3         0.002292   0.099620   0.023  0.98164
## dispersion   0.143111   0.044394   3.224  0.00127 **
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.6479     0.1126   5.754 8.69e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 287.58,  AIC = 585.16
##
## [[44]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.802583   0.053421 108.621 < 2e-16 ***
## ivar2       -0.005223   0.069597  -0.075  0.940179
## ivar3        0.065873   0.063102   1.044  0.296526
## dispersion   0.055558   0.014610   3.803  0.000143 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4356     0.1351   3.223  0.00127 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 290.02,  AIC = 590.04
##
## [[45]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,

```

```

##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.06319    0.07021  72.113 <2e-16 ***
## ivar2        -0.19311    0.07946  -2.430  0.0151 *
## ivar3        -0.09526    0.07431  -1.282  0.1999
## dispersion   0.06913    0.02015   3.431  0.0006 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5459     0.1182   4.619 3.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 254.62, AIC = 519.25
##
## [[46]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.21783    0.08056  64.767 < 2e-16 ***
## ivar2        -0.05815    0.09501  -0.612 0.540546
## ivar3        -0.15474    0.08634  -1.792 0.073120 .
## dispersion   0.10093    0.02876   3.510 0.000448 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5358     0.1307   4.101 4.11e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 271.03, AIC = 552.06
##
## [[47]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05)), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.00880    0.06797  73.686 < 2e-16 ***
## ivar2        -0.08071    0.08762  -0.921  0.357

```

```

## ivar3      -0.12451    0.08257  -1.508    0.132
## dispersion  0.09678    0.02450   3.950 7.83e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4054     0.1359   2.982 0.00286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 263.95, AIC = 537.89
##
## [[48]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.01622    0.16867  29.741 <2e-16 ***
## ivar2        0.06436    0.11446   0.562  0.574
## ivar3        0.12060    0.10673   1.130  0.258
## dispersion   0.19706    0.08604   2.290  0.022 *
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.7739     0.1037   7.466 8.27e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 261.33, AIC = 532.66
##
## [[49]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.61399    0.13549  34.055 < 2e-16 ***
## ivar1        0.01424    0.00423   3.366 0.000761 ***
## ivar2       -0.10125    0.07667  -1.321 0.186627
## ivar3       -0.12346    0.07323  -1.686 0.091827 .
## dispersion   0.07189    0.01892   3.799 0.000145 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4440     0.1341   3.311 0.000929 ***
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.18,  AIC = 524.36
##
## [[50]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.422338   0.080906  67.020 < 2e-16 ***
## ivar1         0.013819   0.002534   5.453 4.95e-08 ***
## ivar2        -0.127150   0.053287  -2.386  0.017 *
## ivar3        -0.070192   0.050375  -1.393  0.163
## dispersion    0.048919   0.010376   4.715 2.42e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.1659     0.1375   1.206  0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 291.6,  AIC = 595.21
##
## [[51]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.874931   0.166888  35.203 < 2e-16 ***
## ivar1        -0.014414   0.005244  -2.749 0.005984 **
## ivar2        -0.137849   0.093496  -1.474 0.140377
## ivar3        -0.021591   0.085914  -0.251 0.801573
## dispersion    0.101176   0.026892   3.762 0.000168 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.5113     0.1185   4.314 1.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 284.73,  AIC = 581.45
##
## [[52]]
##

```

```

## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.087783   0.057413 106.034 < 2e-16 ***
## ivar1        -0.010558   0.001829  -5.773 7.78e-09 ***
## ivar2        -0.022250   0.039134  -0.569  0.5697
## ivar3         0.062361   0.036397   1.713  0.0866 .
## dispersion   0.030851   0.006739   4.578 4.70e-06 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.02806    0.14346   0.196   0.845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 281.29, AIC = 574.58
##
## [[53]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.898841   0.128285  38.187 < 2e-16 ***
## ivar1         0.005758   0.004034   1.428 0.153417
## ivar2        -0.193186   0.073378  -2.633 0.008470 **
## ivar3        -0.095743   0.069408  -1.379 0.167765
## dispersion   0.060803   0.016415   3.704 0.000212 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.4841     0.1297   3.734 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 253.76, AIC = 519.52
##
## [[54]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##

```

```

## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.415639   0.149993  36.106 < 2e-16 ***
## ivar1        -0.007224   0.004741  -1.524 0.127622
## ivar2        -0.061273   0.087572  -0.700 0.484127
## ivar3        -0.166754   0.079791  -2.090 0.036627 *
## dispersion   0.089320   0.024159   3.697 0.000218 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4788     0.1283   3.733 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 269.97, AIC = 551.94
##
## [[55]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.160830   0.130640  39.504 < 2e-16 ***
## ivar1        -0.005543   0.004175  -1.328 0.1842
## ivar2        -0.084544   0.080049  -1.056 0.2909
## ivar3        -0.127852   0.076138  -1.679 0.0931 .
## dispersion   0.087808   0.021460   4.092 4.28e-05 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.3322     0.1320   2.517 0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 263.21, AIC = 538.43
##
## [[56]]
##
## Call:
## gcmr(formula = f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,
##      0), options = list(seed = round(runif(1, 1, 1e+05))), nrep = c(100,
##      1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.562465   0.132752  41.901 < 2e-16 ***
## ivar1        -0.021759   0.004254  -5.115 3.14e-07 ***
## ivar2         0.026777   0.079837   0.335 0.737326
## ivar3         0.082005   0.075945   1.080 0.280234

```

```
## dispersion    0.075901    0.019700    3.853 0.000117 ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.4078      0.1350    3.021 0.00252 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 255.79,  AIC = 523.58
```

Now extract AICs of th 56 models

```
gcmr_aics <- lapply( formula_vec, function(f) {
  fit1 <- gcmr( f, data = bris, marginal = negbin.marg, cormat = arma.cormat(1,0), options = list(seed=ro
  fit1$convergence <- 0
  # fit1$coefficients <- coef( summary(fit1))
  fit1$aic <- AIC(fit1)
  return(fit1$aic)
})
#names(gcmr_res) <- formula_vec

aics_x1 <- as.data.frame(cbind(formula_vec, gcmr_aics))
aics_x1
```

```
##          formula_vec gcmr_aics
## 1      dvar1 ~ ivar1 523.9653
## 2      dvar2 ~ ivar1 597.4623
## 3      dvar3 ~ ivar1 579.5916
## 4      dvar4 ~ ivar1 573.5659
## 5      dvar5 ~ ivar1 523.1739
## 6      dvar6 ~ ivar1 552.3313
## 7      dvar7 ~ ivar1 537.7687
## 8      dvar8 ~ ivar1 520.9003
## 9      dvar1 ~ ivar2 529.8396
## 10     dvar2 ~ ivar2 607.4861
## 11     dvar3 ~ ivar2 583.1568
## 12     dvar4 ~ ivar2 589.1244
## 13     dvar5 ~ ivar2 518.8675
## 14     dvar6 ~ ivar2 553.0775
## 15     dvar7 ~ ivar2 537.9958
## 16     dvar8 ~ ivar2 531.9113
## 17     dvar1 ~ ivar3 528.8248
## 18     dvar2 ~ ivar3 609.2848
## 19     dvar3 ~ ivar3 584.4768
## 20     dvar4 ~ ivar3 588.0453
## 21     dvar5 ~ ivar3 522.889
## 22     dvar6 ~ ivar3 550.42
## 23     dvar7 ~ ivar3 536.697
## 24     dvar8 ~ ivar3 530.9729
## 25     dvar1 ~ ivar1+ivar2 524.8627
## 26     dvar2 ~ ivar1+ivar2 595.0281
## 27     dvar3 ~ ivar1+ivar2 579.5183
## 28     dvar4 ~ ivar1+ivar2 575.3364
## 29     dvar5 ~ ivar1+ivar2 519.375
## 30     dvar6 ~ ivar1+ivar2 553.8645
```



```
## 31      dvar7 ~ ivar1+ivar2  539.0118
## 32      dvar8 ~ ivar1+ivar2  522.7872
## 33      dvar1 ~ ivar1+ivar3  523.9053
## 34      dvar2 ~ ivar1+ivar3  598.3098
## 35      dvar3 ~ ivar1+ivar3  581.5605
## 36      dvar4 ~ ivar1+ivar3  572.8936
## 37      dvar5 ~ ivar1+ivar3   523.723
## 38      dvar6 ~ ivar1+ivar3  550.4167
## 39      dvar7 ~ ivar1+ivar3  537.4728
## 40      dvar8 ~ ivar1+ivar3   521.699
## 41      dvar1 ~ ivar2+ivar3  530.0508
## 42      dvar2 ~ ivar2+ivar3  608.9144
## 43      dvar3 ~ ivar2+ivar3  585.1579
## 44      dvar4 ~ ivar2+ivar3  590.0415
## 45      dvar5 ~ ivar2+ivar3  519.2506
## 46      dvar6 ~ ivar2+ivar3  552.0538
## 47      dvar7 ~ ivar2+ivar3  537.8927
## 48      dvar8 ~ ivar2+ivar3  532.6539
## 49 dvar1 ~ ivar1+ivar2+ivar3  524.3574
## 50 dvar2 ~ ivar1+ivar2+ivar3  595.2073
## 51 dvar3 ~ ivar1+ivar2+ivar3  581.4527
## 52 dvar4 ~ ivar1+ivar2+ivar3   574.576
## 53 dvar5 ~ ivar1+ivar2+ivar3  519.5248
## 54 dvar6 ~ ivar1+ivar2+ivar3  551.9412
## 55 dvar7 ~ ivar1+ivar2+ivar3  538.4233
## 56 dvar8 ~ ivar1+ivar2+ivar3  523.5817
```

Let us choose a model w.r.t each occupation.

```
aic1 <- bind_rows(aics_x1[1,],aics_x1[9,],aics_x1[17,],aics_x1[25,],aics_x1[33,],aics_x1[41,],aics_x1[49,])
aic1
```

```
##          formula_vec gcmr_aics
## 1      dvar1 ~ ivar1  523.9653
## 2      dvar1 ~ ivar2  529.8396
## 3      dvar1 ~ ivar3  528.8248
## 4      dvar1 ~ ivar1+ivar2  524.8627
## 5      dvar1 ~ ivar1+ivar3  523.9053
## 6      dvar1 ~ ivar2+ivar3  530.0508
## 7 dvar1 ~ ivar1+ivar2+ivar3  524.3574
```

```
aic1 <-rownames(aic1)[which.min(aic1[,2])]
aic2 <- bind_rows(aics_x1[2,],aics_x1[10,],aics_x1[18,],aics_x1[26,], aics_x1[34,],aics_x1[42,],aics_x1[50,])
aic2
```

```
##          formula_vec gcmr_aics
## 1      dvar2 ~ ivar1  597.4623
## 2      dvar2 ~ ivar2  607.4861
## 3      dvar2 ~ ivar3  609.2848
## 4      dvar2 ~ ivar1+ivar2  595.0281
## 5      dvar2 ~ ivar1+ivar3  598.3098
## 6      dvar2 ~ ivar2+ivar3  608.9144
## 7 dvar2 ~ ivar1+ivar2+ivar3  595.2073
```

```
aic2 <- rownames(aic2)[which.min(aic2[,2])]
aic3 <- bind_rows(aics_x1[3,],aics_x1[11,],aics_x1[19,],aics_x1[27,], aics_x1[35,],aics_x1[43,],aics_x1[51,])
aic3
```

```
##          formula_vec gcmr_aics
## 1      dvar3 ~ ivar1  579.5916
## 2      dvar3 ~ ivar2  583.1568
## 3      dvar3 ~ ivar3  584.4768
## 4      dvar3 ~ ivar1+ivar2 579.5183
## 5      dvar3 ~ ivar1+ivar3 581.5605
## 6      dvar3 ~ ivar2+ivar3 585.1579
## 7 dvar3 ~ ivar1+ivar2+ivar3 581.4527
```

```
aic3 <- rownames(aic3)[which.min(aic3[,2])]
```

```
aic4 <- bind_rows(aics_x1[4,],aics_x1[12,],aics_x1[20,],aics_x1[28,], aics_x1[36,],aics_x1[44,],aics_x1
aic4
```

```
##          formula_vec gcmr_aics
## 1      dvar4 ~ ivar1  573.5659
## 2      dvar4 ~ ivar2  589.1244
## 3      dvar4 ~ ivar3  588.0453
## 4      dvar4 ~ ivar1+ivar2 575.3364
## 5      dvar4 ~ ivar1+ivar3 572.8936
## 6      dvar4 ~ ivar2+ivar3 590.0415
## 7 dvar4 ~ ivar1+ivar2+ivar3 574.576
```

```
aic4 <- rownames(aic4)[which.min(aic4[,2])]
```

```
aic5 <- bind_rows(aics_x1[5,],aics_x1[13,],aics_x1[21,],aics_x1[29,], aics_x1[37,],aics_x1[45,],aics_x1
aic5
```

```
##          formula_vec gcmr_aics
## 1      dvar5 ~ ivar1  523.1739
## 2      dvar5 ~ ivar2  518.8675
## 3      dvar5 ~ ivar3  522.889
## 4      dvar5 ~ ivar1+ivar2 519.375
## 5      dvar5 ~ ivar1+ivar3 523.723
## 6      dvar5 ~ ivar2+ivar3 519.2506
## 7 dvar5 ~ ivar1+ivar2+ivar3 519.5248
```

```
aic5 <- rownames(aic5)[which.min(aic5[,2])]
```

```
aic6 <- bind_rows(aics_x1[6,],aics_x1[14,],aics_x1[22,],aics_x1[30,], aics_x1[38,],aics_x1[46,],aics_x1
aic6
```

```
##          formula_vec gcmr_aics
## 1      dvar6 ~ ivar1  552.3313
## 2      dvar6 ~ ivar2  553.0775
## 3      dvar6 ~ ivar3   550.42
## 4      dvar6 ~ ivar1+ivar2 553.8645
## 5      dvar6 ~ ivar1+ivar3 550.4167
## 6      dvar6 ~ ivar2+ivar3 552.0538
## 7 dvar6 ~ ivar1+ivar2+ivar3 551.9412
```

```
aic6 <- rownames(aic6)[which.min(aic6[,2])]
```

```
aic7 <- bind_rows(aics_x1[7,],aics_x1[15,],aics_x1[23,],aics_x1[31,], aics_x1[39,],aics_x1[47,],aics_x1
aic7
```

```
##          formula_vec gcmr_aics
## 1      dvar7 ~ ivar1  537.7687
## 2      dvar7 ~ ivar2  537.9958
## 3      dvar7 ~ ivar3   536.697
## 4      dvar7 ~ ivar1+ivar2 539.0118
## 5      dvar7 ~ ivar1+ivar3 537.4728
```

```
## 6      dvar7 ~ ivar2+ivar3  537.8927
## 7 dvar7 ~ ivar1+ivar2+ivar3  538.4233
```

```
aic7 <- rownames(aic7)[which.min(aic7[,2])]
aic8 <- bind_rows(aics_x1[8,],aics_x1[16,],aics_x1[24,],aics_x1[32,], aics_x1[40,],aics_x1[48,],aics_x1[56,],aics_x1[64,])
aic8
```

```
##          formula_vec gcmr_aics
## 1      dvar8 ~ ivar1  520.9003
## 2      dvar8 ~ ivar2  531.9113
## 3      dvar8 ~ ivar3  530.9729
## 4      dvar8 ~ ivar1+ivar2  522.7872
## 5      dvar8 ~ ivar1+ivar3   521.699
## 6      dvar8 ~ ivar2+ivar3  532.6539
## 7 dvar8 ~ ivar1+ivar2+ivar3  523.5817
```

```
aic8 <- rownames(aic8)[which.min(aic8[,2])]
aic_best <- c(aic1,aic2,aic3,aic4,aic5,aic6,aic7,aic8)
aic_best
```

```
## [1] "5" "4" "4" "5" "2" "5" "3" "1"
```

The information for the best feature set is

```
## [1] "5" "4" "4" "5" "2" "5" "3" "1"
```

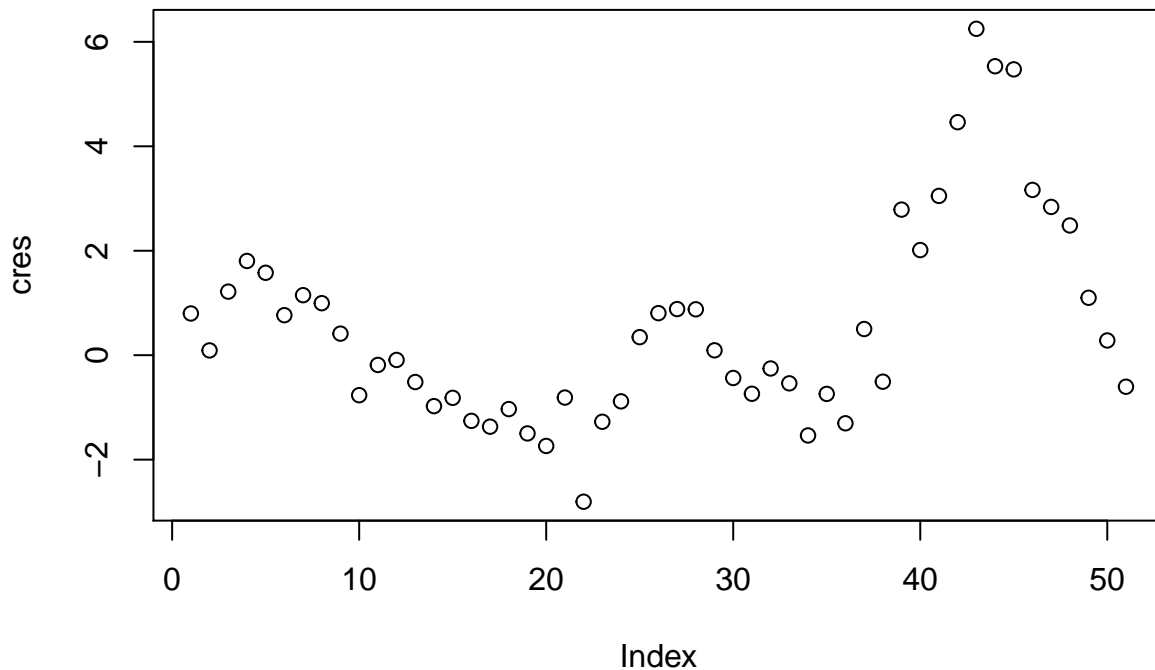
For job 1, 4 and 6, the best model is  $xi \sim Mth.No + \sin(Mth.No * 2 * \pi/12)$ . For job 2 and 3, the best model is  $xi \sim Mth.No + \cos(Mth.No * 2 * \pi/12)$ . For job 5, the best model is  $xi \sim Mth.No + \cos(Mth.No * 2 * \pi/12)$ . For job 7, the best model is  $x7 \sim Mth.No + \sin(Mth.No * 2 * \pi/12)$ .

First we look at occupation 1 - Managers, we fit a Gaussian copula marginal regression with two terms based on the AIC criterion. Extract the residuals, take cumulative sum.

```
fit1 <- gcmr(x1 ~ Mth.No + sin(Mth.No * 2 * pi/12), data = bris, marginal=negbin.marg, cormat = arma.cormat(1, 0))
fit1$convergence <- 0
summary(fit1)
```

```
##
## Call:
## gcmr(formula = x1 ~ Mth.No + sin(Mth.No * 2 * pi/12), data = bris,
##       marginal = negbin.marg, cormat = arma.cormat(1, 0), options = list(seed = round(runif(1,
##       1, 1e+05))), nrep = c(100, 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.62380    0.14500  31.887 < 2e-16 ***
## Mth.No           0.01411    0.00453   3.116 0.001835 **
## sin(Mth.No * 2 * pi/12) -0.11725    0.07843  -1.495 0.134930
## dispersion       0.07881    0.02181   3.613 0.000302 ***
##
## Coefficients Gaussian copula:
##              Estimate Std. Error z value Pr(>|z|)
## ar1      0.4922      0.1261    3.903 9.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 256.95, AIC = 523.9
```

```
res <- resid(fit1)
cres<- cumsum(res)
plot(cres)
```

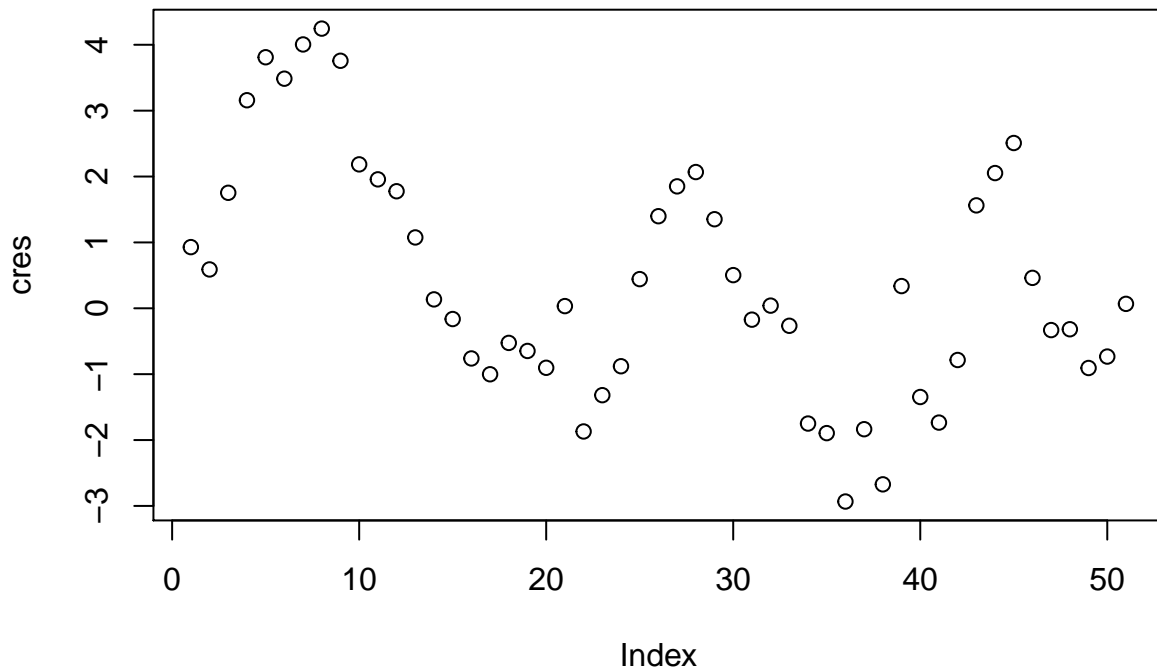


There is a lack of consistency in the temporal trend. Moreover, the change point occur at month 42 when the formerly increasing trend faltens and changes to decreasing in Brisbane. Hence we will incorporate an extra interaction term and refit gcmr.

```
fit1 <- gcmr(x1 ~ Mth.No + sin(Mth.No * 2 * pi/12) + Mth.No*as.factor(Mth.No < 42), data = bris, marginal = negbin.marg,
fit1$convergence <- 0
summary(fit1)
```

```
##
## Call:
## gcmr(formula = x1 ~ Mth.No + sin(Mth.No * 2 * pi/12) + Mth.No *
##       as.factor(Mth.No < 42), data = bris, marginal = negbin.marg,
##       cormat = arma.cormat(1, 0), options = list(seed = round(runif(1,
##       1, 1e+05))), nrep = c(100, 1000), no.se = FALSE, opt = myopt))
##
##
## Coefficients marginal model:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    10.126234    1.108482   9.135 < 2e-16
## Mth.No         -0.103164    0.023195  -4.448 8.68e-06
## sin(Mth.No * 2 * pi/12) -0.042562    0.049265  -0.864  0.388
## as.factor(Mth.No < 42)TRUE -5.574986    1.103704  -5.051 4.39e-07
## Mth.No:as.factor(Mth.No < 42)TRUE  0.120521    0.022934   5.255 1.48e-07
## dispersion      0.044443    0.009965   4.460 8.19e-06
##
## (Intercept)      ***
## Mth.No           ***
## sin(Mth.No * 2 * pi/12)
## as.factor(Mth.No < 42)TRUE      ***
```

```
## Mth.No:as.factor(Mth.No < 42)TRUE ***
## dispersion ***
##
## Coefficients Gaussian copula:
##      Estimate Std. Error z value Pr(>|z|)
## ar1 0.005809   0.165540   0.035   0.972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## log likelihood = 250.45, AIC = 514.9
res <- resid(fit1)
cres<- cumsum(res)
plot(cres)
```



We obtained mathematically, the fitted model of job counts  $X_t$  in month  $t$  is given by  $X_t | \mathcal{F}_{t-1} \sim \text{NB}(\mu_t)$  with

$$\ln(\mu_t) = 10.13 - 0.10t + 0.12I_{(t < 42)}.$$

## Reference

Akaike, Hirotugu. 1998. “Information Theory and an Extension of the Maximum Likelihood Principle.” In *Selected Papers of Hirotugu Akaike*, 199–213. New York, NY: Springer New York. [https://doi.org/10.1007/978-1-4612-1694-0\\_15](https://doi.org/10.1007/978-1-4612-1694-0_15).

Krainski, Elias T., Virgilio Gómez-Rubio, Haakon Bakka, Amanda Lenzi, Daniela Castro-Camilo, Daniel Simpson, Finn Lindgren, and Havard Rue. n.d. “Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and Inla.” <https://becarioprecario.bitbucket.io/spde-gitbook/>.

Liboschik, Tobias. 2016. “Modeling Count Time Series Following Generalised Linear Models.” PhD thesis, Department of Statistics, TU Dortmund University.

Lindgren, Finn, and Havard Rue. n.d. “Bayesian Spatial Modelling with R-Inla.” <https://www.stat.washington.edu/peter/591/Lindgren.pdf>.

Nelson, Roger. 2006. “An Introduction to Copulas.” *Springer Series in Statistics*. Springer New York.  
<https://doi.org/10.1007/0-387-28678-0>.