Journal of Statistical Software

April 2016, Volume 70, Code Snippet 1. *doi: 10.18637/jss.v070.c01*

Generating Correlated and/or Overdispersed Count Data: A SAS Implementation

George Kalema
Makerere University

Geert Molenberghs
Universiteit Hasselt

Abstract

Analysis of longitudinal count data has, for long, been done using a generalized linear mixed model (GLMM), in its Poisson-normal version, to account for correlation by specifying normal random effects. Univariate counts are often handled with the negative-binomial (NEGBIN) model taking into account overdispersion by use of gamma random effects. Inherently though, longitudinal count data commonly exhibit both features of correlation and overdispersion simultaneously, necessitating analysis methodology that can account for both. The introduction of the combined model (CM) by Molenberghs, Verbeke, and Demétrio (2007) and Molenberghs, Verbeke, Demétrio, and Vieira (2010) serves this purpose, not only for count data but for the general exponential family of distributions. Here, a Poisson model is specified as the parent distribution of the data with a normally distributed random effect at the subject or cluster level and/or a gamma distribution at observation level. The GLMM and NEGBIN model are special cases. Data can be simulated from (1) the general CM, with random effects, or, (2) its marginal version directly. This paper discusses an implementation of (1) in SAS software (SAS Inc. 2011). One needs to reflect on the mean of both the combined (hierarchical) and marginal models in order to generate correlated and/or overdispersed counts. A pre-specification of the desired marginal mean (in terms of covariates and marginal parameters), a marginal variance-covariance structure and the hierarchical mean (in terms of covariates and regression parameters) is required. The implied hierarchical parameters, the variance-covariance matrix of the random effects, and the variance-covariance matrix of the overdispersion part are then derived from which correlated Poisson data are generated. Sample calls of the SAS macro are presented as well as output.

Keywords: copulas, correlated data, multivariate gamma distribution, Poisson distribution.

1. Introduction

In many areas of research, correlated or otherwise hierarchical data are collected, be it in medical statistical, social, behavioral, and educational science, economy, biology, agriculture,

etc. Molenberghs and Verbeke (2005) and Verbeke and Molenberghs (2000) describe methods for the analysis of discrete and continuous longitudinal data, respectively. Apart from analyzing data, leading to estimation and hypothesis testing, interest may be in evaluating the statistical properties of models fitted to data generated from certain mechanisms, to study operational characteristics, for design and sample size determination purposes, etc. Monte-Carlo (MC) simulation is a commonly followed route to this effect. Simulation of hierarchical normally distributed data is relatively straightforward, because simulation from univariate and multivariate normal distributions is easy.

Our focus is on hierarchical and/or overdispersed count data. A common basic model is the Poisson model. If overdispersion is present, i.e., the variance is unequal to the mean, the negative binomial (NEGBIN) model can be considered. This model can be generated by assuming that the Poisson parameter follows a gamma distribution. For correlated counts, the generalized linear mixed model (GLMM; Breslow and Clayton 1993) can be used. Here, the linear predictor of the Poisson model encompasses normal random effects. When both phenomena are present, the so-called combined model (CM) is relevant (Molenberghs *et al.* 2007, 2010); it allows for both normal and gamma random effects. Consider a longitudinal study, where a subject is measured repeatedly over time. The normal random effects induce correlation between different measures of the same subject. gamma random effects at each occasion take care of overdispersion. Should the gamma random effects be correlated, then a serial correlation process can be modeled. This parallels the flexibility of the general linear mixed models (Verbeke and Molenberghs 2000). Two comments are in place. First, the GLMM and CM allow for general outcome types, so there are versions for binary and time-to-event outcomes; these are not the focus here. Second, the GLMM and CM are often called conditional models, because they are formulated conditional on random effects. Marginal functions, namely, the mean and variance, are then derived by marginalizing/integrating over the random effects. While marginalizing over the normal random effects is relatively easy for count data, the expressions are more cumbersome than marginalizing over correlated or uncorrelated gamma random effects. This feature is exploited by offering implementations of the CM in a SAS 9.3 (SAS Inc. 2011) macro to generate correlated and/or overdispersed Poisson random variables. The method can also be used to generate purely serially correlated counts by dropping the normal random effects and choosing the "overdispersion part" to follow a serially correlated multivariate gamma distribution. The macro makes use of the SAS `LOGISTIC` procedure to create the design matrix by using a working response variable, which is deleted after it has served its purpose. At a data manipulation phase, the macro uses the `GENMOD` procedure to obtain parameter estimates corresponding to the design matrix, to eliminate columns from the design matrix corresponding to reference categories (when dummy coding is used for some covariates).

Proposals have been made in the literature. Examples include the overlapping sums (Madsen and Dalthorp 2007; Mardia 1970; Kocherlakota and Kocherlakota 1992, 2001); lognormal-Poisson hierarchy; a method named "normal-to-anything" (NORTA; Cario and Nelson 1997; Cario and Nelson 1998; Nelsen 2006; Mardia 1970; Li and Hammond 1975), and extensions thereof (Yahav and Shmueli 2012; Ghosh and Pasupathy 2012; Shin and Pasupathy 2010; Avramidis, Channouf, and L'Ecuyer 2009; Park and Shin 1998; Downer and Moser 2001). Next, we briefly describe these methods in turn. Overlapping sums (also known as the trivariate reduction) method generates a bivariate Poisson random vector from three or more independent Poisson random variables. NORTA presents a way of generating corre-

lated random variables derived from the multivariate normal distribution given pre-specified marginals and a desired correlation structure. This is motivated by the ease with which multivariate normal random variables can be generated. A log-normal-Poisson hierarchy generates correlated counts by transforming correlated normals to log-normal variables, such that the two random variables are correlated via the correlation in the conditional means. See also Devroye (1986) for an overview on random variate generation. Most of these methods suffer from such limitations as: severe computational restrictions; difficulty achieving the target correlation; generated variables are required to be overdispersed; correlations constrained to be strictly positive; etc. These issues stem form the fact that the multivariate Poisson distribution grows in computational complexity with an increase in the dimensions due to the summations inherent in the distribution (Karlis 2003).

A review of the models used for correlated and/or overdispersed count data is presented in Section 2, while Section 3 focuses on data generation. Section 4 provides some specific details about the SAS macro, some examples of CMs to generate data from, how to generate these data using the macro and the output of the macro. Some concluding remarks are given in Section 5.

2. Overview of the models

We seek to generate correlated Poisson random variables for K independent subjects with subject i having measurements Y_{ij}, $i = 1, \ldots, K$, $j = 1, \ldots, n_i$. Our methodology requires a reflection on both the target marginal mean model as well as the hierarchical model. The target or desired mean must be pre-specified in terms of an $n_i \times p$ known design matrix $\widetilde{\boldsymbol{X}}_i$ and a corresponding p-dimensional vector of parameters for the marginal mean $\boldsymbol{\alpha}$. In addition, the covariates for the hierarchical component, i.e., conditional on the normal random effects, also need to be specified from which design matrices $\boldsymbol{X}_i$ of dimension $n_i \times m$ and $\boldsymbol{Z}_i$, an $n_i \times q$ design matrix of subject i are created. An m-dimensional fixed-effects parameter vector $\boldsymbol{\beta}$, D the normal random effects variance-covariance matrix of dimension $q \times q$ and $\boldsymbol{\Sigma}_i$, the $n_i \times n_i$ variance-covariance matrix of the gamma random effects, are then derived and used to generate the data. Details of these are given next.

In dealing with correlated count data, the so-called generalized linear mixed model (GLMM; Breslow and Clayton 1993; Wolfinger and O'Connell 1993; Molenberghs and Verbeke 2005) is a commonly used tool for analysis. The GLMM modifies the linear predictor in generalized linear models (GLM; Nelder and Wedderburn 1972; McCullagh and Nelder 1989; Agresti 2002), a class of fixed-effects models unifying linear, logistic, and Poisson regression models among others, to include unknown subject-specific or random effects in addition to the fixed effects. In practice, these random effects are usually assumed to follow a normal distribution, mainly for convenience and software availability. However, they can, in principle, be assumed to follow a different distribution than the normal. Specific to count data, the standard GLMM (also known as the Poisson-normal model) takes the following form:

$$\begin{aligned} Y_{ij}|\boldsymbol{b}_i &\sim \text{Poi}(\lambda_{ij}), \\ \ln(\lambda_{ij}) &= \boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{Z}_{ij}^\top \boldsymbol{b}_i, \\ \boldsymbol{b}_i &\sim \text{N}(0, D), \end{aligned} \tag{1}$$

whereby the conditional distribution of the observations from a subject i given the random

effects $\boldsymbol{b}_i$ is Poisson with a rate parameter λ_{ij} that is log-linearly related to covariates.

Count data are also notorious for overdispersion (Hinde and Demétrio 1998a,b; Breslow 1984; Lawless 1987; Molenberghs and Verbeke 2005), a feature that is usually accounted for by using the NEGBIN model. The NEGBIN model follows from a two-stage approach where in stage 1, a distribution is considered for the response variable, given a random effect $f(\boldsymbol{y}_i|\boldsymbol{b}_i)$, and in stage 2, a model for the random effects $f(\boldsymbol{b}_i)$ is specified, in this case the Poisson and gamma distributions, respectively. Combining the two stages and integrating as in Equation 2 over the random effects results in the NEGBIN marginal model. Extension to the case of correlated or hierarchical count data is rather easy as shown in Section 3.2 of Molenberghs *et al.* (2010). Fitting these models is done by maximizing the marginal likelihood

$$f(\boldsymbol{y}_i) = \int f(\boldsymbol{y}_i|\boldsymbol{b}_i) f(\boldsymbol{b}_i) d\boldsymbol{b}_i. \tag{2}$$

Closed-form expressions for these integrals do not exist in all cases but Molenberghs *et al.* (2007) and Molenberghs *et al.* (2010) derived the marginal mean and covariance for the Poisson GLMM case as

$$\mathsf{E}(Y_{ij}) = \mu_{ij} = \ln(\lambda_{ij}) = \boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + 0.5 \boldsymbol{Z}_{ij}^\top D \boldsymbol{Z}_{ij}, \tag{3a}$$

$$\mathsf{VAR}(\boldsymbol{Y}_i) = \boldsymbol{M}_i + \boldsymbol{M}_i \left(e^{\boldsymbol{Z}_i D \boldsymbol{Z}_i^\top} - \boldsymbol{J}_i \right) \boldsymbol{M}_i, \tag{3b}$$

respectively, where $\boldsymbol{J}_i$ is a matrix of 1's and $\boldsymbol{M}_i$ is a diagonal matrix with entries μ_{ij}. Also, the higher-order marginal moments and the marginal joint distribution can be derived in closed form for the Poisson case (Molenberghs *et al.* 2010).

Much as longitudinal count data usually exhibit both features of correlation and overdispersion, analysis until recently has been done predominantly by accounting only for one of these features but not both. The introduction of the CM by Booth, Casella, Friedl, and Hobert (2003), Molenberghs *et al.* (2007) and Molenberghs *et al.* (2010) quite flexibly accounts for these features simultaneously. The CM is expressed as

$$Y_{ij} \sim \text{Poi}(\lambda_{ij}^*), \tag{4a}$$

$$\lambda_{ij}^* = \theta_{ij} \lambda_{ij} = \theta_{ij} \exp\left(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{Z}_{ij}^\top \boldsymbol{b}_i \right), \tag{4b}$$

$$\boldsymbol{\theta}_i \sim \text{MGamma}(\text{mean} = \boldsymbol{1}, \text{variance} = \boldsymbol{\Sigma}_i), \tag{4c}$$

$$\boldsymbol{b}_i \sim \text{N}(0, D), \tag{4d}$$

where θ_{ij}, the entries in $\boldsymbol{\theta}_i$, are the overdispersion parameters introduced at observation level. If the θ_{ij}'s are assumed to be independent as is often done in practice, then the association is only induced by the $\boldsymbol{b}_i$ and the θ_{ij} would cover the overdispersion not accounted for by the normal random effects. As such, $\boldsymbol{\Sigma}_i$ is reduced to a diagonal matrix. On the other hand, the θ_{ij} can be correlated such that $\boldsymbol{\Sigma}_i$ can take on more general structures, which implies the use of some form of multivariate gamma (MGamma) distribution. The marginal mean and the marginal variance-covariance matrix take the form:

$$\mathsf{E}(Y_{ij}) = \mu_{ij} = \theta_{ij} \exp\left(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + 0.5 \boldsymbol{Z}_{ij}^\top D \boldsymbol{Z}_{ij} \right), \tag{5a}$$

$$\mathsf{VAR}(\boldsymbol{Y}_i) = \boldsymbol{V}_i = \boldsymbol{M}_i + \boldsymbol{M}_i (\boldsymbol{P}_i - \boldsymbol{J}_i) \boldsymbol{M}_i, \tag{5b}$$

where $\boldsymbol{M}_i = \text{diag}(\boldsymbol{\mu}_i)$ and

$$\boldsymbol{P}_i = e^{\left(0.5\boldsymbol{Z}_i D \boldsymbol{Z}_i^\top\right)} \left(\boldsymbol{\Sigma}_i + \boldsymbol{J}_i\right) e^{\left(0.5\boldsymbol{Z}_i D \boldsymbol{Z}_i^\top\right)}.$$

Here, $\boldsymbol{J}_i$ is a matrix of ones. Note that we make use of the fact that the Gamma random effects have unit mean. Note also that the overall variance-covariance matrix $\boldsymbol{V}_i$ is made up, among others, of the normal random-effects variance D and the Gamma random-effects variance $\boldsymbol{\Sigma}_i$. The rules for creating $\boldsymbol{Z}_i$ follow those of $\boldsymbol{X}_i$, given that both are design matrices.

3. Generation of correlated counts

As will be presented in Section 3.1, the marginal moments of the GLMM can be used to parsimoniously generate correlated count data with a pre-specified marginal mean function and such variance-covariance structures as compound symmetry and the one generated by random intercept and random slope models. In the GLMM case, however, the random effects used do not distinguish between correlation and overdispersion, a disadvantage that may lead to mis-representation of the random-effects variability and therefore necessitates the CM. Since marginal and conditional parameters do not have a 1:1 correspondence in the case of discrete data, unlike the continuous data case, caution has to be exercised when generating data for the context in mind, marginal or hierarchical. The use of random effects in the hierarchical context is satisfying while a marginal model or moments thereof are necessary in the marginal context in order to match context and process. We describe the process of count data generation, first, starting from a GLMM and then from a CM. A presentation of the algorithms for generating data from these two models follows in Sections 3.1 and 3.2.

3.1. The GLMM as a data generator

The GLMM can be used to generate correlated random variables with a desired structure. Given a marginal (log) mean, possibly depending on covariates $\widetilde{\boldsymbol{X}}_i$, and a variance-covariance matrix for $\boldsymbol{Y}_i$, Algorithm 1 below generates random variables with this pre-specified structure.

Algorithm 1:

1. *Derive the unknowns $\boldsymbol{\beta}$ and D of the GLMM by comparing the desired marginals $\boldsymbol{\mu}_i$ and $\boldsymbol{V}_i$ with the marginals (3a) and (3b) from the GLMM.*
2. *Using D, simulate $\boldsymbol{b}_i$.*
3. *Compute* $\ln(\lambda_{ij}) = \boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{Z}_{ij}^\top \boldsymbol{b}_i$.
4. *Simulate* $Y_{ij} \sim \text{Poi}(\lambda_{ij})$.

For example, in the case of compound symmetry (CS) and given the desired marginal mean as $\ln(\boldsymbol{\mu}_i) = \widetilde{\boldsymbol{X}}_i \boldsymbol{\alpha}$ and desired variance-covariance structure as $\boldsymbol{V}_i = \boldsymbol{M}_i + \tau^2 \boldsymbol{J}_i$ (CS structure), the necessary unknowns in Step 1 of the above algorithm are derived by comparing [a] $\widetilde{\boldsymbol{X}}_i \boldsymbol{\alpha} = \boldsymbol{X}_i \boldsymbol{\beta} + 0.5 \boldsymbol{Z}_i D \boldsymbol{Z}_i^\top$ [which is (3a) expressed in matrix form] for the marginal mean, and, [b] $\boldsymbol{M}_i + \tau^2 \boldsymbol{J}_i = \boldsymbol{M}_i + \boldsymbol{M}_i \left(e^{\boldsymbol{Z}_i D \boldsymbol{Z}_i^\top} - \boldsymbol{J}_i\right) \boldsymbol{M}_i$ for the marginal variance-covariance structure. The

marginal mean parameter $\boldsymbol{\alpha}$ was introduced in Section 2. Note that $\widetilde{\boldsymbol{X}}_i$ is the marginal design matrix, as opposed to $\boldsymbol{X}_i$, which is the design matrix for the fixed effects given the normal random effects. Solving [a] for $\boldsymbol{\beta}$ and [b] for D leads to:

$$\boldsymbol{\beta} = \left(\boldsymbol{X}_i^\top \boldsymbol{X}_i\right)^- \boldsymbol{X}_i^\top \left(\widetilde{\boldsymbol{X}}_i \boldsymbol{\alpha} - 0.5 \boldsymbol{Z}_i D \boldsymbol{Z}_i^\top\right), \tag{6a}$$

$$D = \left(\boldsymbol{Z}_i^\top \boldsymbol{Z}_i\right)^- \boldsymbol{Z}_i^\top \log\left(\boldsymbol{M}_i^{-1} \tau^2 \boldsymbol{J}_i \boldsymbol{M}_i^{-1} + \boldsymbol{J}_i\right) \boldsymbol{Z}_i \left(\boldsymbol{Z}_i^\top \boldsymbol{Z}_i\right)^-, \tag{6b}$$

where $(.)^-$ indicates a generalized inverse. For a general $\boldsymbol{V}_i$, $\tau^2 \boldsymbol{J}_i$ in (6b) becomes $\boldsymbol{V}_i - \boldsymbol{M}_i$. Then, it follows that $\mathsf{E}(\boldsymbol{Y}_i) = e^{\widetilde{\boldsymbol{X}}_i \boldsymbol{\alpha}}$ and $\mathsf{VAR}(\boldsymbol{Y}_i) = \boldsymbol{V}_i$. If the generalized inverse is not an inverse, the solution clearly is not unique. This is not a problem; it simply means that several choices of $\boldsymbol{\beta}$ and D are possible, which nevertheless all lead to the desired marginal structure. This is akin to the fact that there is a one-to-many map between a given marginal model on the one hand and the class of hierarchical models that marginalizes to it on the other. Any member of the class of hierarchical model can in principle be used as a data generator for the marginal structure.

3.2. The combined model as a data generator

While the GLMM is relatively standard, it allows for specific covariance and correlation structures only, i.e., these that are induced from the normal random effects. In contrast, the CM can be used to generate correlated Poisson random variables, with very general marginal covariance and correlation structures. This generality was conceptually presented in Molenberghs *et al.* (2007) and Molenberghs *et al.* (2010), but not used in practice. Precisely, the earlier paper was restricted to i.i.d. gamma variables, whereas here, they can follow any multivariate gamma distribution. The major extension relative to the GLMM is that there is a third unknown term in the CM, i.e., $\boldsymbol{\Sigma}_i$, the variance-covariance matrix for the overdispersion parameter(s). It allows for positive and negative correlations alike, provided the resulting marginal variance-covariance matrix V_i is positive-definite. However, because this matrix is user-specified in our algorithms, checking for this property is trivial.

Given a desired mean and variance-covariance structure, Algorithm 2 generates the Poisson variates.

Algorithm 2:

1. *Derive the unknowns* $\boldsymbol{\beta}$, D, *and* $\boldsymbol{\Sigma}_i$ *in the CM.*
2. *Generate* $\boldsymbol{\theta}_i \sim \text{MGamma}(mean = \mathbf{1}, variance = \boldsymbol{\Sigma}_i)$.
3. *Using* D, *simulate* $\boldsymbol{b}_i$.
4. *Compute* $\lambda_{ij}^* = \theta_{ij} \exp(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{Z}_{ij}^\top \boldsymbol{b}_i)$.
5. *Simulate* $Y_{ij} \sim \text{Poi}(\lambda_{ij}^*)$.

The necessary unknowns in Step 1 of Algorithm 2 are given by $\boldsymbol{\beta}$ as in (6a) and further:

$$D = \left(\boldsymbol{Z}_i^\top \boldsymbol{Z}_i\right)^- \boldsymbol{Z}_i^\top \log\left[\boldsymbol{M}_i^{-1}(\boldsymbol{V}_i - \boldsymbol{M}_i)\boldsymbol{M}_i^{-1} + \boldsymbol{J}_i\right] \boldsymbol{Z}_i \left(\boldsymbol{Z}_i^\top \boldsymbol{Z}_i\right)^-,$$

$$\boldsymbol{\Sigma}_i = e^{-\boldsymbol{Z}_i D \boldsymbol{Z}_i^\top} \left[\boldsymbol{M}_i^{-1}(\boldsymbol{V}_i - \boldsymbol{M}_i)\boldsymbol{M}_i^{-1} + \boldsymbol{J}_i\right] - \boldsymbol{J}_i,$$

			Gamma random effects		
			Yes		No
			Correlated	Independent	
Normal random effects	SAS macro argument and corresponding input		`GammaRandEff = 2`	`GammaRandEff = 1`	`GammaRandEff = 0`
Yes	Correlated	`NormalRandEff = 2`	✓	✓	✓
Yes	Independent	`NormalRandEff = 1`	✓	✓	✓
No		`NormalRandEff = 0`	✓	✓	✗

Table 1: Possible combinations of the normal and gamma random effects in the context of count data. ✓ refers to combinations of the CM from which correlated and/or overdispersed data can be generated, while ✗ refers to the independent count data generation case which is not of interest in this paper. Also included is the SAS macro argument referring to the normal and gamma random effects and the corresponding value to be input for each case, when running macro `CorrPoisson`.

where notational conventions are as before.

An extension to generating purely serially correlated outcomes is done by removing the normal random effect and choosing $\boldsymbol{\theta}_i$ such that it follows a serially correlated multivariate gamma.

The general form of the CM (4), in the case of Poisson data, is that the normal random effects are correlated and the gamma random effects are also correlated. From this general case, several special cases can be derived. An overview of the possible combinations is presented in Table 1. The following special cases, which are also presented in Table 1, can be derived from the more general case:

- A combination of normal and independent gamma random effects. This is the most commonly used form of the CM in which the normal random effects induce/account for correlation while the gamma random effects induce/account for overdispersion. It is model (4) but with $\boldsymbol{\Sigma}_i$ diagonal.

- Normal random effects without gamma random effects. In this case, (4) reduces to (1) and data are generated as explained in Section 3.1. Here, the normal random effects induce/account for both correlation and overdispersion.

- No normal random effects, no gamma random effects. The absence of both random effects is equivalent to generating independent counts.

- No normal random effects, correlated gamma random effects. This implies that both correlation and overdispersion are induced via the gamma random effects. Thus, λ_{ij} in (4) becomes $\exp(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta})$ and $\boldsymbol{\Sigma}_i$ is fully general. This case allows for arbitrary marginal variance-covariance matrices between the Poisson variables, featuring positive and/or negative correlation, provided the resulting variance-covariance matrix is positive-definite.

- No normal random effects, independent gamma random effects. In this case, the CM reduces to the negative-binomial model which accounts for overdispersion but not correlation. Then, λ_{ij} in (4) becomes $\exp(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta})$ and $\boldsymbol{\Sigma}_i$ is diagonal.

Additional variations can be made by choosing for the normal random effects (random intercept + slope, or higher dimensions) to be either independent (D diagonal) or correlated.

4. The SAS macro

4.1. Introduction

We have implemented the above discussed method of data generation for marginal models in SAS version 9.3. The SAS macro is called `CorrPoisson`. Data generation is done in SAS/IML preceded by some data manipulations. Given that we allow the θ_{ij}'s to be correlated, some form of multivariate gamma distribution is required. We invoke the **copula** package (Hofert, Kojadinovic, Maechler, and Yan 2015; Hofert and Maechler 2011; Yan 2007; Kojadinovic and Yan 2010) available for the R environment for statistical computing and graphics (R Core Team 2016) to do this. We are aware of the experimental `COPULA` procedure in SAS 9.3 but follow a different route here. Instead, we have explored SAS 9.3's flexibility to call R in the SAS/IML procedure using the `SUBMIT` and `ENDSUBMIT` statements. This is done in the `RinSASIML.sas` program which is included in `CorrPoisson.sas` using the `%inc` statement. When calling `CorrPoisson`, the path to `RinSASIML.sas` must be defined correctly using the `Include` macro argument as, for example, `Include = c:/temp` in case `RinSASIML.sas` is saved in the `c:/temp` directory. Three issues deserve mentioning at this point, namely, (a) that R software (available from `https://CRAN.R-project.org`) has to be installed as well as the **copula** package (it is not installed by default in R), (b) that the SAS system has to be launched with the `-RLANG` system option to permit calling R from SAS (note that it is often convenient to insert this option in a `SASV9.CFG` file), and, (c) that calling R functions in SAS using the `SUBMIT` and `ENDSUBMIT` statements is a relatively new feature that was introduced in SAS/IML 9.22. As such, the macro will not work with SAS versions that preceded 9.22. We refer to the SAS/IML 9.22 user's guide (SAS Inc. 2011) or later versions for details about calling R from within SAS. The macro was developed and certainly works with SAS version 9.3 and R version 2.15.2. However, compatibility issues may arise while invoking R in SAS/IML depending on the versions of both R and SAS. Error messages that may indicate incompatibility are, for example, `An installed version of R could not be found` or `The installed version of R cannot be used`. Table 2 (obtained from the blog available at Wicklin 2013) presents an overview of the match between the latest SAS versions and the corresponding R releases they support. In general, specific SAS versions support specific sets of R releases. It is necessary that the user first ensures that calling R from SAS is permitted. A quick test is to run the following code in SAS:

SAS Version	`PROC IML`	SAS/IML Studio	Release Date	R Versions
9.2	N/A	3.2	Jul 2009	2.6.1 – 2.11.1
9.22	9.22	3.3	Nov 2010	2.9.1 – 2.11.1
9.3	9.3	3.4	Jul 2011	2.9.1 – 2.15.3
9.3m2	12.1	12.1	Aug 2012	2.9.1 – 2.15.3
9.4	12.3	12.3	Jul 2013	2.13.0 – 3.0.1
9.4m1	13.1	13.1	Dec 2013	2.13.0 – present

Table 2: SAS compatibility with R releases as obtained from Wicklin (2013)'s blog.

```
proc options option=RLANG;
proc iml;
submit /R;
getwd()
endsubmit;
run;
```

If this test results in errors and the user has the most recent version of R, it may be useful to explicitly tell SAS/IML which R version to use since SAS/IML tries to use the corresponding `R_HOME` variable. This can be done by launching SAS with a specification of the `R_HOME` variable as, for example, `-RLANG -SET R_HOME "C:/program files/R/R-3.0.1"`, in which case R version 3.0.1 would be used. We emphasize that in order to have a successful execution of macro `CorrPoisson`, one should first ensure that (1) the connection between SAS and R is ok (by running the test program above), and (2) that the path to `RinSASIML.sas` is correctly defined. Otherwise, errors directly related to these two aspects may be encountered. Note that our macro can quite easily use any parameterization method possible in SAS `LOGISTIC` and `GENMOD` procedures for the design matrix for classification variables, e.g., effect, glm, ordinal, reference, etc. See SAS documentation about the different parameterization methods for classification variables at http://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_logistic_sect006.htm. In what follows, in Sections 4.2 and 4.3, we illustrate how to use the macro with two examples, namely, (1) when a random intercept model is specified for the normal random effects and (2) when a random intercept and slope model is used for the normal random effects. Please note that while the methodology and the SAS macro allow for $\widetilde{\boldsymbol{X}}_i$ to be different from $\boldsymbol{X}_i$, our illustrations below set $\widetilde{\boldsymbol{X}}_i = \boldsymbol{X}_i$, by leaving the macro argument `Xcov2` blank, for purposes of checking whether or not the data generation follows intuition. A need to have $\widetilde{\boldsymbol{X}}_i$ and $\boldsymbol{X}_i$ different is achieved by specifying the different covariates in the macro arguments `Xcov` and `Xcov2`, respectively. A general remark about the use of the macro is that all variables to be specified in the different macro arguments must be in the dataset specified in the `CovData` argument. A detailed description of the macro arguments is given in Tables 3 and 4.

4.2. The random intercept case

In this section, we demonstrate how to generate correlated count data from the CM using the macro that we have developed.

The combined model

Using a random intercept model for the normal random effects, we illustrate the generation of 4 correlated Poisson variables using the following CM:

$$\begin{aligned} Y_{ij} &\sim \text{Poi}(\lambda^*_{ij}), \\ \lambda^*_{ij} = \theta_{ij}\lambda_{ij} &= \theta_{ij}\exp(\beta_0 + b_{0i} + \beta_1 T_i + \beta_2 t_{ij} + \beta_3 T_i * t_{ij}), \\ \boldsymbol{\theta}_i &\sim \text{MGamma}(\text{mean} = \mathbf{1}, \text{variance} = \boldsymbol{\Sigma}_i), \\ \boldsymbol{b}_i = b_{0i} &\sim \text{N}(0, d), \end{aligned} \tag{7}$$

where, b_{0i} is the random intercept, the treatment allocation $T_i \sim$ Bernoulli(0.5), t_{ij} is the ordering of the jth observation in subject $i = 1, \ldots, K = 500$ and $j = 1, 2, 3, 4$. The design

matrices $\boldsymbol{X}_i$ and $\widetilde{\boldsymbol{X}}_i$ in (6a) are created from the same covariates, namely, T_i, t_{ij}, and $T_i t_{ij}$ while the desired marginal mean parameters and desired variance-covariance structure are

$$\boldsymbol{\alpha} = \begin{pmatrix} 1.521 \\ 0.437 \\ -0.254 \\ 0.145 \end{pmatrix}$$

and

$$\boldsymbol{V}_i = \begin{pmatrix} 256 & 128 & 144 & 224 \\ 128 & 208 & 228 & 172 \\ 144 & 228 & 299 & 296 \\ 224 & 172 & 296 & 567 \end{pmatrix},$$

respectively.

Calling macro `CorrPoisson`

The CM has several variations, as shown in Table 1 and described in Section 3.2. Generating data from these variations can be achieved by specifying the combinations of normal and gamma random effects using the macro arguments `NormalRandEff` and `GammaRandEff`, respectively. For illustration purposes, we shall only generate from the general case of the CM, namely, with correlated normal and correlated gamma random effects. This is done by specifying `NormalRandEff = 2` and `GammaRandEff = 2`. More specifically, to generate data from CM (7), one has to first create the `CovData` dataset, referred to as `temp` in our example. This dataset contains variables `id, trt` and `time`. Here, `trt` stands for "treatment". Further, `time` indicates time in the study, either from a well-chosen baseline moment 0, recoded to be centered around 0, or otherwise depending on the study. See Tables 3 and 4 for more details about the `CovData` argument. Given `temp`, the following macro call would generate correlated count data from the CM with a random intercept model specified for the normal random effects:

```
%CorrPoisson(CovData = temp, id = id, OrderVar = time,
  Xcov = trt time trt*time, Alpha = 1.521 0.437 -0.254 0.145,
  Class = trt, outData = out, random =,
  desiredVarCov = 256 128 144 224 208 228 172 299 296 567,
  GammaRandEff = 2, NormalRandEff = 2);
```

Generating from the other variations is done by specifying the macro arguments as shown in Tables 1, 3 and 4. Please note that setting `NormalRandEff = 0` meaning no normal random effects, and `GammaRandEff = 0` meaning no gamma random effects, would imply generating independent count data which is not of interest in this paper although it is also implemented in macro `CorrPoisson`, for completeness. Also note that not all macro arguments are shown in the above call. Those not shown are set to the default values. See Tables 3 and 4 for the full list of macro arguments that facilitate the data generation and their descriptions. Leaving the `random` argument blank implies the use of a random intercept model for the normal random effects.

Argument	Description
`CovData`	Dataset containing subject or cluster identification variable and covariates from which design matrices $\widetilde{\boldsymbol{X}}_i$ (and $\boldsymbol{X}_i$) in (6a) is (are) created. By default, `CovData = temp`, for illustration purposes. Dataset `temp` contains covariates `id, trt` and `time`. It is a required argument and therefore has to be specified before running the macro. This dataset should take the "long" or hierarchical data structure as opposed to the wide format. Please note that all variables to be specified in the other macro arguments must be in this dataset.
`ID`	Subject identification variable in `CovData`.
`OrderVar`	Variable with ordering of the observations within subject. This should be contained in the `CovData` dataset. Default input is `time`, again for illustrating how the macro can be invoked.
`Xcov`	Covariates from which $\widetilde{\boldsymbol{X}}_i$ is created. It is also a required argument though for illustration purposes, `Xcov = trt time trt*time`. Please note that the intercept must not be included in the specification of `Xcov`. It is added at the creation of the $\widetilde{\boldsymbol{X}}_i$ design matrix.
`Xcov2`	Covariates from which $\boldsymbol{X}_i$ is created. If left blank (the default), the macro sets `Xcov2 = Xcov` such that $\boldsymbol{X}_i = \widetilde{\boldsymbol{X}}_i$ thus using the same covariates for both these design matrices. Again, the intercept must not be included in the specification of `Xcov2`. It is added at the creation of the $\boldsymbol{X}_i$ design matrix if different covariates from those in `Xcov` are specified.
`Alpha`	The desired marginal mean parameter estimates, without the reference categories. It can be specified, for example, as `Alpha = 2.5 0.7 1.2 -0.45`, corresponding to `Intercept`, `trt=0`, `time` and `trt*time`, respectively. Please note that the parameter estimate for the intercept must always be included.
`Class`	Specify all classification variables included in the `Xcov` argument, for example, `Class = trt`.
`Class2`	Specify all classification variables included in the `Xcov2` argument, for example, `Class = gender`. Note that if `Xcov2` is left blank, this argument is ignored.
`outdata`	Name of final output dataset in which the generated outcome variable, named `Y`, is merged with the `CovData` dataset. It also contains the variance-covariance matrix of the gamma distribution ($\boldsymbol{\Sigma}_i$, `GammaCov`) in (4c), the corresponding correlation matrix of the gamma distribution (`GammaCorr`), the `shape` and `scale` parameters for the gamma distribution, the θ_i's (`GamV`) in (4c), the λ_{ij}^*'s (`mu`) in 4b and the random effects estimates ($\boldsymbol{b}_i$) in (4d). By default, `outdata = out`.

Table 3: The macro arguments for `CorrPoisson` and their corresponding description, part I.

Output

By default, `CorrPoisson` creates the output dataset `out` whose content is as described in Tables 3 and 4 under the `outData` argument and provided in a SAS dataset called `out1.sas7bdat` for the random intercept case. By setting `Estimates = 1` (the default), the following output

Argument	Description
`param`	Parameterization method for the classification variables specified in the `Class` argument. Default is `param = glm`. See SAS documentation for details about the different methods at http://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_logistic_sect006.htm.
`random`	Covariates for the normal random effects. Default is blank hence random intercept model. To fit, for example, a random intercept and slope model, specify `random = time`, whereby `time` indicates the ordering of observations within a subject.
`meanNormalRE`	By default (`meanNormalRE` is left blank), macro `CorrPoisson` then assumes the normal random effects ($\boldsymbol{b}_i$) to have zero mean. This can be changed by filling the mean in this argument.
`seed`	Set seed. Default is `seed = 123`.
`desiredVarCov`	Specify the desired variance-covariance matrix $\boldsymbol{V}_i$ by inputting the entries of the upper triangular matrix row-wise (e.g., v11 v12 v22 or v11 v12 v13 v22 v23 v33).
`GammaRandEff`	Either `0`, `1` or `2` for the gamma random effects where `0` = No gamma random effects, `1` = Independent gamma random effects, `2` = Correlated gamma random effects. Default is `GammaRandEff = 2`.
`NormalRandEff`	Either `0`, `1` or `2` for the normal random effects where `0` = No normal random effects, `1` = Independent normal random effects, `2` = Correlated normal random effects. Default is `NormalRandEff = 2`.
`Estimates`	Print results to output window, "Yes" (`Estimates = 1`) or "No" (`Estimates = 0`). Default is `Estimates = 1`.
`Include`	Specifies the path to `RinSASIML.sas`. It is a required argument and must be correctly specified for the macro to run well.

Table 4: The macro arguments for `CorrPoisson` and their corresponding description, part II.

is generally printed to the output window:

1. The CM combination of normal and gamma random effects from which data is being generated.
2. The sample size or number of clusters (K) in the `CovData` dataset.
3. Minimum and maximum number of measurements per cluster or subject.
4. Covariates used for the design matrix $\boldsymbol{Z}_i$ of the normal random effects (including the intercept). For example, if a random intercept model is considered for the normal random effects, then "Normal random effects covariates = Intercept". In the case of a random intercept and slope model, then "Normal random effects covariates = Intercept time" where time is the variable specified in the macro argument `OrderVar` in Tables 3 and 4.
5. The desired or given marginal mean in terms of the covariates and parameters. For classification variables, e.g., `trt` in this case, an indication is given for the reference category being considered. Herein, 0 was considered as the reference leading to "trt0".

```
   Generate Correlated Poisson data from CM with
     Normal and Correlated Gamma random effects
*****************************************************

              Sample size (K) = 500
   minimum number of measurements per subject = 4
   maximum number of measurements per subject = 4
    Normal random effects covariates = Intercept


 Given Mean parameters are: Intercept =     1.521
                            trt0            0.437
                            time           -0.254
                            trt0time        0.145

                                      Y1        Y2        Y3        Y4

Given variance-covariance matrix =  Y1       256       128       144       224
                                    Y2       128       208       228       172
                                    Y3       144       228       299       296
                                    Y4       224       172       296       567


  Parameter      alpha       beta       diff          D

  Intercept      1.521   1.518115   0.002885    0.00577
  trt0           0.437      0.437  -3.13E-14
  time          -0.254     -0.254   1.61E-15
  trt0time       0.145      0.145  -3.47E-15
```

Figure 1: Results printed by macro `CorrPoisson` to the output window when a random intercept model is used for the normal random effects.

6. The desired or given variance-covariance matrix $\boldsymbol{V}_i$.

7. The covariates used, the desired marginal mean parameters ("alpha"), the derived conditional parameters ("beta"), the difference between the marginal and conditional parameters ("diff") and the variance covariance matrix D of the normal random effects. Please note that D is printed only if normal random effects are used.

Specifically, Figure 1 shows the results printed to the output window when a random intercept model is used for the normal random effects. From Figure 1 and as expected for a CM with a random intercept only model, a change (diff) in the marginal parameters ($\boldsymbol{\alpha}$) from the hierarchical parameters ($\boldsymbol{\beta}$) is only in the intercept parameter but the other parameters remain practically unchanged.

To illustrate the functionality of the method given a low variance-covariance structure, the following macro call was used (for all the macro arguments, see Tables 3 and 4):

```
%CorrPoisson(CovData = temp, id = id, OrderVar = time,
  Xcov = trt time trt*time, Alpha = -1.021 0.837 -0.254 0.845, Class = trt,
  outData = out1smallvar, random =, desiredVarCov = 0.25 0.20 0.25,
  GammaRandEff = 0, NormalRandEff = 1);
```

```
   Generate Correlated Poisson data from CM with
         Normal and No Gamma random effects
*************************************************

              Sample size (K) = 20
   minimum number of measurements per subject = 2
   maximum number of measurements per subject = 2
    Normal random effects covariates = Intercept


 Given Mean parameters are: Intercept =    -1.021
                            trt0           0.837
                            time          -0.254
                            trt0time       0.845


                                          Y1        Y2

Given variance-covariance matrix =  Y1      0.25       0.2
                                    Y2       0.2      0.25


  Parameter     alpha      beta      diff         D

  Intercept    -1.021 -1.028092 0.0070918 0.0141835
  trt0          0.837     0.837 5.551E-15
  time         -0.254    -0.254 1.832E-15
  trt0time      0.845     0.845 -2.11E-15
```

Figure 2: Results printed by macro `CorrPoisson` to the output window when a random intercept model is used given a small variance-covariance structure.

For very small variance-covariance matrix $\boldsymbol{V}_i$, computational issues are more rampant as the method gets more sensitive also to the given $\boldsymbol{\alpha}$ parameters. Figure 2 shows the output as printed to the output window. The output dataset is also presented in a SAS dataset named `out1smallvar.sas7bdat`.

4.3. The random intercept and slope case

Consider the following CM with a random intercept and slope model specified for the normal random effects:

$$\begin{aligned} Y_{ij} &\sim \text{Poi}(\lambda^*_{ij}), \\ \lambda^*_{ij} = \theta_{ij}\lambda_{ij} &= \theta_{ij}\exp((\beta_0 + b_{0i}) + \beta_1 T_i + (\beta_2 + b_{1i})t_{ij} + \beta_3 T_i * t_{ij}), \\ \boldsymbol{\theta}_i &\sim \text{MGamma}(\text{mean} = \mathbf{1}, \text{variance} = \boldsymbol{\Sigma}_i), \\ \boldsymbol{b}_i = \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} &\sim \text{N}\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, D = \begin{pmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{pmatrix}\right], \end{aligned} \tag{8}$$

where notation and the desired marginals are as in Section 4.2. Since model (8) is similar to (7) with the only difference being the random slope (b_{1i}) in model (8), the code shown

```
   Generate Correlated Poisson data from CM with
Correlated Normal and Correlated Gamma random effects
 ****************************************************

                Sample size (K) = 500
   minimum number of measurements per subject = 4
   maximum number of measurements per subject = 4
 Normal random effects covariates = Intercept time


  Given Mean parameters are: Intercept =     1.521
                             trt0            0.437
                             time           -0.254
                             trt0time        0.145

                                        Y1        Y2        Y3        Y4

Given variance-covariance matrix =  Y1       256       128       144       224
                                    Y2       128       208       228       172
                                    Y3       144       228       299       296
                                    Y4       224       172       296       567


      Parameter     alpha      beta      diff         D

      Intercept     1.521 1.5195865 0.0014135 0.0040014 0.0000601
      trt0          0.437     0.437 -1.59E-14 0.0000601 0.0002349
      time         -0.254 -0.254647 0.0006473
      trt0time      0.145     0.145 -1.86E-15
```

Figure 3: Results printed by macro `CorrPoisson` to the output window when a random intercept and slope model is used for the normal random effects and when the subjects have equal number of measurements.

in Section 4.2 is used to generate correlated count data. The only difference is that, in this case, the argument `random = time`. The macro will then create a dataset containing the generated data named `out`, in long form, and a print to the output window as in Figure 3. Unlike the random intercept case, the difference between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the case of the random intercept and slope model is evident in both the intercept and time parameters, as expected. However, this pattern holds when all subjects in the `CovData` dataset have an equal number of measurements, when the random intercept and slope model for the normal random effects is specified. Note that for the random intercept model, a change is seen only in the intercept parameter, whether or not all subjects have an equal number of measurements. Not only does the macro allow for n_i to be equal but it also allows n_i to be unequal. To illustrate this, we generate a dataset with n_i being between 2 and 4 measurements per subject. For one to fit the same model shown in Section 4.3 but with n_i varying between 2 and 4 measurements, the same code as in the above call would be used. The difference, though, should be in the input dataset `temp` specified using the `CovData` argument. The results as printed to the output window are shown in Figure 4. One should note the difference in the two models from the minimum and maximum number of measurements per subject in the output.

```
   Generate Correlated Poisson data from CM with
Correlated Normal and Correlated Gamma random effects
 ****************************************************

                Sample size (K) = 500
   minimum number of measurements per subject = 2
   maximum number of measurements per subject = 4
 Normal random effects covariates = Intercept time


  Given Mean parameters are: Intercept =     1.521
                             trt0            0.437
                             time           -0.254
                             trt0time        0.145


                                        Y1        Y2        Y3        Y4

Given variance-covariance matrix =  Y1       256       128       144       224
                                    Y2       128       208       228       172
                                    Y3       144       228       299       296
                                    Y4       224       172       296       567


     Parameter      alpha      beta      diff         D

     Intercept      1.521 1.5209283 0.0000717 0.0071187 -0.001993
     trt0           0.437 0.4370443 -0.000044 -0.001993 0.0016015
     time          -0.254 -0.255718 0.0017181
     trt0time       0.145 0.1449747 0.0000253
```

Figure 4: Results printed by macro `CorrPoisson` to the output window when a random intercept and slope model is used for the normal random effects and there are varying number of measurements per subject.

5. Concluding remarks

We have presented SAS code to generate correlated Poisson data, in the context of marginal models, from the CM introduced by Molenberghs *et al.* (2007) and Molenberghs *et al.* (2010). The CM simultaneously accommodates correlation and overdispersion unexplained by the normal random effects. In the absence of correlation, the model simplifies to a negative-binomial model for overdispersion. On the other hand, in the absence of overdispersion, it simplifies to the GLMM. The model's flexible structure makes it a good candidate as a data generator reflecting the characteristics of interest, in this case, overdispersion and/or correlation. The CM is a convenient tool that mimics or incorporates these intrinsic features of correlated count data.

By marginalizing the distribution of the Poisson response conditional on the normal and gamma random effects, i.e., integrating out the random effects from the conditional density of the CM, one is able to generate data in the context of marginal models by comparing the mean and variance of the marginal model with the desired marginal mean and variance structures. The covariates determining the fixed- and random-effects design matrices are kept simple herein. This is not limiting in the sense that a specification of any covariates can be

done as is needed. It is possible to encounter non-positive definite D matrices or negative entries along the diagonal of $\mathbf{\Sigma}_i$. These issues are important but different. A non-positive D matrix points to a non-allowable hierarchical model. However, the corresponding marginal model may then still be valid. If interest is restricted to marginal quantities, such a model may be retained. A non-valid $\mathbf{\Sigma}_i$ is a problem in any case. Thus, keeping the inferential goals in mind, such model settings need to be given careful consideration. A purely marginal specification comes with the advantage that a broader parameter space is allowable. At the same time, a purely marginal specification inhibits the use of such hierarchical features as empirical Bayes predictions.

Because the CM is hierarchical, random variables with only positive correlations are generated due to restrictions of positive-definiteness on the random effects variance-covariance matrices. This may be a drawback for the CM, as is the case for some of the methods present in the literature for count data generation. However, a way to overcome this is to generate directly from the marginal model, arguably via correlated θ_{ij}, of which the variance-covariance matrix $\mathbf{\Sigma}_i$ then reflects the desired structure.

Execution errors of the form `Unable to allocate sufficient memory` may be encountered when attempts are made to generate sizes of datasets that use resources more than are available to SAS for the specific computer in use. One may then have to reduce the sample size or the number of measurements per subject or find a computer with greater memory capacity. Our experience is that this limitation is more rampant in 32-bit Windows operating systems which support matrices up to a maximum size of memory of 2GB of addressable space.

The SAS macro `CorrPoisson.sas` and the `RinSASIML.sas` program are available as supplementary material along with this manuscript and at the authors' website (http://ibiostat.be/software/overdispersion).

Acknowledgments

The authors gratefully acknowledge support from IAP Research Network P7/06 of the Belgian Government (Belgian Science Policy).

References

Agresti A (2002). *Categorical Data Analysis*. 2nd edition. John Wiley & Sons, New York.

Avramidis AN, Channouf N, L'Ecuyer P (2009). "Efficient Correlation Matching for Fitting Discrete Multivariate Distributions with Arbitrary Marginals and Normal-Copula Dependence." *INFORMS Journal on Computing*, **2**, 88–106. doi:10.1287/ijoc.1080.0281.

Booth JG, Casella G, Friedl H, Hobert JP (2003). "Negative Binomial Loglinear Mixed Models." *Statistical Modelling*, **3**, 179–181. doi:10.1191/1471082x03st058oa.

Breslow N (1984). "Extra-Poisson Variation in Log-Linear Models." *Applied Statistics*, **33**, 38–44. doi:10.2307/2347661.

Breslow N, Clayton D (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **88**, 9–25. doi:10.1080/01621459.1993.10594284.

Cario MC, Nelson BL (1997). "Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix." *Technical report*, Northwestern University, Evanston, Illinois.

Cario MC, Nelson BL (1998). "Numerical Methods for Fitting and Simulating Autoregressive-to-Anything Processes." *INFORMS Journal on Computing*, **10**, 72–81. doi:10.1287/ijoc.10.1.72.

Devroye L (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York. doi:10.1007/978-1-4613-8643-8.

Downer R, Moser E (2001). "On the Generation of a Multivariate Spatial Poisson Distribution." *Technical report*, Lousiana State University.

Ghosh S, Pasupathy R (2012). "C-NORTA: A Rejection Procedure for Sampling from the Tail of Bivariate NORTA Distributions." *INFORMS Journal on Computing*, **24**, 295–310. doi:10.1287/ijoc.1100.0447.

Hinde J, Demétrio CGB (1998a). "Overdispersion: Models and Estimation." *Computational Statistics & Data Analysis*, **27**, 151–170. doi:10.1016/s0167-9473(98)00007-3.

Hinde J, Demétrio CGB (1998b). "Overdispersion: Models and Estimation." *Technical report*, São Paulo.

Hofert M, Kojadinovic I, Maechler M, Yan J (2015). ***copula**: Multivariate Dependence with Copulas*. R Foundation for Statistical Computing. R package version 0.999-14, URL https://CRAN.R-project.org/package=copula.

Hofert M, Maechler M (2011). "Nested Archimedean Copulas Meet R: The **nacopula** Package." *Journal of Statistical Software*, **39**(9), 1–20. doi:10.18637/jss.v039.i09.

Karlis D (2003). "An EM Algorithm for Multivariate Poisson Distribution and Related Models." *Journal of Applied Statistics*, **30**, 63–77. doi:10.1080/0266476022000018510.

Kocherlakota S, Kocherlakota K (1992). *Bivariate Discrete Distributions*. CRC Press, Boca Raton.

Kocherlakota S, Kocherlakota K (2001). "Regression in the Bivariate Poisson Distribution." *Communications in Statistics, Theory & Methods*, **30**, 815–825. doi:10.1081/sta-100002259.

Kojadinovic I, Yan J (2010). "Modeling Multivariate Distributions with Continuous Margins Using the **copula** R Package." *Journal of Statistical Software*, **34**(9), 1–20. doi:10.18637/jss.v034.i09.

Lawless JF (1987). "Negative Binomial and Mixed Poisson Regression." *The Canadian Journal of Statistics*, **15**, 209–225. doi:10.2307/3314912.

Li ST, Hammond JL (1975). "Generation of Pseudo-Random Numbers with Specified Univariate Distributions and Correlation Coefficients." *IEEE Transactions on Systems, Man, and Cybernetics*, **5**, 557–561. doi:10.1109/tsmc.1975.5408380.

Madsen L, Dalthorp D (2007). "Simulating Correlated Count Data." *Environmental and Ecological Statistics*, **14**, 129–148. doi:10.1007/s10651-007-0008-1.

Mardia KV (1970). *Families of Bivariate Distributions*. Griffin, London.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman & Hall, London. doi:10.1007/978-1-4899-3242-6.

Molenberghs G, Verbeke G (2005). *Models for Discrete Longitudinal Data*. Springer-Verlag, New York. doi:10.1007/978-1-4419-0300-6.

Molenberghs G, Verbeke G, Demétrio C (2007). "An Extended Random Effects Approach to Modeling Repeated Overdispersed Count Data." *Lifetime Data Analysis*, **13**, 513–531. doi:10.1007/s10985-007-9064-y.

Molenberghs G, Verbeke G, Demétrio C, Vieira A (2010). "A Family of Generalized Linear Models for Repeated Measures with Normal and Conjugate Random Effects." *Statistical Science*, **25**, 325–347. doi:10.1214/10-sts328.

Nelder JA, Wedderburn RWM (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society B*, **135**, 370–384. doi:10.2307/2344614.

Nelsen RB (2006). *An Introduction to Copulas*. Springer-Verlag, Berlin. doi:10.1007/0-387-28678-0.

Park CG, Shin DW (1998). "An Algorithm for Generating Correlated Random Variables in a Class of Infinitely Divisible Distributions." *Journal of Statistical Computation and Simulation*, **61**, 127–139. doi:10.1080/00949659808811905.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

SAS Inc (2011). *SAS/IML 9.3 User's Guide*. Cary, NC. URL http://www.sas.com/.

Shin K, Pasupathy R (2010). "An Algorithm for Fast Generation of Bivariate Poisson Random Vectors." *INFORMS Journal on Computing*, **22**, 81–92. doi:10.1287/ijoc.1090.0332.

Verbeke G, Molenberghs G (2000). *Linear Mixed Models for Longitudinal Data*. Springer-Verlag, New York. doi:10.1007/978-1-4419-0300-6.

Wicklin R (2013). "What Versions of R Are Supported by SAS?" URL http://blogs.sas.com/content/iml/2013/09/16/what-versions-of-r-are-supported-by-sas/.

Wolfinger R, O'Connell M (1993). "Generalized Linear Mixed Models: A Pseudo-Likelihood Approach." *Journal of Statistical Computing and Simulation*, **48**, 233–243. doi:10.1080/00949659308811554.

Yahav I, Shmueli G (2012). "On Generating Multivariate Poisson Data in Management Science Applications." *Applied Stochastic Models for Business and Industry*, **28**, 91–102. doi:10.1002/asmb.901.

Yan J (2007). "Enjoy the Joy of Copulas: With a Package **copula**." *Journal of Statistical Software*, **21**(4), 1–21. doi:10.18637/jss.v021.i04.

Affiliation:

George Kalema
School of Statistics and Applied Economics
Makerere University
P.O. Box 7062
Kampala, Uganda
and
Interuniversity Institute for Biostatistics and statistical Bioinformatics (I-BioStat)
Universiteit Hasselt
Agoralaan 1
B3590 Diepenbeek, Belgium
E-mail: george.kalema@uhasselt.be
URL: http://ibiostat.be/

Geert Molenberghs
Interuniversity Institute for Biostatistics and statistical Bioinformatics (I-BioStat)
Universiteit Hasselt
Agoralaan 1
B3590 Diepenbeek, Belgium
and
KU Leuven
Kapucijnenvoer 35
B3000 Leuven, Belgium
E-mail: geert.molenberghs@uhasselt.be
URL: http://ibiostat.be/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
April 2016, Volume 70, Code Snippet 1 *Submitted:* 2013-08-27
doi:10.18637/jss.v070.c01 *Accepted:* 2015-04-10