A Copula Approach to Generate Non-Normal Multivariate Data for SEM

Patrick Mair and Albert Satorra and Peter M. Bentler

Original Citation:

Mair, Patrick and Satorra, Albert and Bentler, Peter M.
(2011)
A Copula Approach to Generate Non-Normal Multivariate Data for SEM.
This version is available at: https://epub.wu.ac.at/3122/
Available in ePubWU: June 2011

epubWU, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.
A Copula Approach to Generate Non-Normal Multivariate Data for SEM

Patrick Mair
Institute for Statistics and Mathematics
WU Vienna University of Economics and Business

Albert Satorra
Department of Economics and Business
Universitat Pompeu Fabra and Barcelona GSE

Peter M. Bentler
Department of Psychology and Statistics
University of California, Los Angeles
Abstract

The present paper develops a procedure based on multivariate copulas for simulating multivariate non-normal data that satisfies a pre-specified covariance matrix. The covariance matrix used, can comply with a specific moment structure form (e.g., a factor analysis or a general SEM model). So the method is particularly useful for Monte Carlo evaluation of SEM models in the context of non-normal data. The new procedure for non-normal data simulation is theoretically described and also implemented on the widely used R environment. The quality of the method is assessed by performing Monte Carlo simulations. Within this context a one-sample test on the observed VC-matrix is involved. This test is robust against normality violations. This test is defined through a particular SEM setting. Finally, an example for Monte Carlo evaluation of SEM modeling of non-normal data using this method is presented.

Keywords: Multivariate Copulas, Structural Equation Modeling (SEM), Robust One-Sample Test Covariance Matrix, Monte Carlo Simulations, Non-Normal Data SEM.
1 Introduction

Examining the Structural Equation Model (SEM) literature with respect to Monte Carlo simulations that involves the generation of multivariate non-normal distributions, the data generating process is either related to steer the skewness and kurtosis (Curran et al., 1996; Finch et al., 1997; Mattson, 1997; Reinartz et al., 2002; Lei and Lomax, 2005), or to create contaminated normals (Chen and Portnoy, 1996; Yuan and Bentler, 2007; Mair et al., 2010).

Obviously, there are many kinds of univariate probability distributions, but only in a few cases there is a native multivariate analogue. A very flexible tool to approach this multivariate distribution problem is provided by copulas (see, e.g., Nelsen, 2006) which help understanding relationships among multivariate outcomes. Basically, a copula is a function that links univariate marginals to a joint multivariate distribution.

Over recent years copulas have become popular in fields like finance, insurance, risk management, and econometrics. Typically, in practical applications, researchers observe a multivariate outcome. Using copulas, this multivariate outcome can be described and, consequently, conclusions regarding the univariate margins can be drawn. To our knowledge, copulas are not yet widely used in the fields of behavioral and social sciences, as well as psychometrics.

In this article we propose a data generation approach for non-normal multivariate data based on copulas that can be used, among others, for subsequent SEM simulation studies. An important task in SEM simulation studies is to examine the behavior of parameter estimates and fit statistics for non-normal multivariate data. The crucial issue in such data generation processes is that the simulated data need to obey a certain model or true variance-covariance (VC) structure.

Simulating multinormal data with a corresponding VC-structure is trivial, nowadays. Most of statistical software packages include a module for multinormal data simulations. Simulating non-normal multivariate data is still a challenging task. Especially within the framework of SEM, since we want to simulate the data in a systematic manner by posing a certain covariance structure on them. Within the next sections we give some basic copula theory, discuss various types of copulas, present a two-stage data generation approach, show how the data simulation can be performed in R (R Development Core Team, 2011), and give a SEM Monte Carlo example.

2 Copula theory

The starting point for our copula elaborations is the inversion method (also known as inversion transformation sampling) for generating random samples from any probability distribution given its cumulative distribution function (cdf; see e.g. Hörmann et al., 2004, p. 14).

**Theorem 2.1** Let \( F(x) \) be a continuous cdf and \( U \) a uniform \( U(0, 1) \) random number. The random variable \( X = F^{-1}(U) \) has the cdf \( F \).

In simple words this implies that we can create random numbers for any continuous, univariate distribution with cdf \( F \) by sampling from a uniform distribution. Copulas are essentially the generalization of this principle to multivariate distributions. They were introduced by Sklar (1959) by the following theorem (which is now known as Sklar’s theorem):
Theorem 2.2  Given a joint distribution function \( F(x_1, \ldots, x_d) \) for random variables \( X_1, \ldots, X_d \) with marginals \( F_1(x_1), \ldots, F_n(x_d) \), \( F \) can be written as a function of its marginals:

\[
F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_n(x_d)),
\]

where \( C(u_1, \ldots, u_d) \) is a joint distribution with uniform marginals. If each \( F_i \), with \( i = 1, \ldots, d \), is continuous, \( C \) is unique. \( C \) is called a **copula** and is a multivariate distribution with the property that marginals are all uniform over \([0, 1]\).

Theorem 2.3  Assume that each \( F_i \) and \( C \) are differentiable. The joint density \( f(x_1, \ldots, x_d) \) can be written as

\[
f(x_1, \ldots, x_d) = f_1(x_1) \times \cdots \times f_n(x_n)c(F_1(x_1), \ldots, F_d(x_d)),
\]

where \( f_i(x_i) \) is the (continuous) density corresponding to the distribution function \( F_i(x_i) \), and \( c = \partial^n C / (\partial F_1 \cdots \partial F_d) \) is called the copula density.

Essentially, these two results are sufficient to construct a multivariate distribution based on the copula. We see that the copula is a function of the marginals that it allows the “coupling” of the marginals into a multivariate joint distribution. Within a simulation context this implies that, first, we draw random variables from known continuous univariate distributions (e.g., normal, uniform, Weibull, etc.). Second, we create a copula that captures the dependence among the random variables involving corresponding dependency parameters.

As we know, SEM is a multivariate covariance-based method. Within a simulation context, we have to take into account the following issues:

1. allow for a multivariate specification (some copulas are only defined for the bivariate case),
2. have a corresponding \( d \times d \) correlation (or covariance) matrix as parameter (in addition to the parameters for the univariate margins), and
3. allow for arbitrary, continuous univariate margins.

Over the years, many types of copulas have been proposed (see Nelsen, 2006, for an overview). Obviously, since SEM is a multivariate method, only multivariate copulas are relevant for our data generation process. The following subsections give an overview of such types of copulas. In the next section we describe the most popular copula, the Gaussian copula, in more detail. Then we elaborate other types of copulas relevant for our approach of data generation where the data should reflect a particular VC-structure.

### 2.1 Gaussian copula

The elaborations in this section are inspired by the articles of Clemen and Reilly (1999) and Schözel and Friederichs (2008) which give a very simple introduction to multivariate normal copulas. Basically, normal or Gaussian copulas can be derived by a simple back and forth transformation of the random variables to the multivariate standard normal distribution, as we will show below. A normal copula captures dependencies in the same way as the multivariate normal distribution does, i.e., using pairwise correlations among the variables. The difference is that it does so for variables with arbitrary marginals.
In order to show the construction of a Gaussian copula we start with normal distributed random variables \( Z_1, \ldots, Z_d \). The univariate normal densities are denoted by \( \phi(z_1), \ldots, \phi(z_d) \) and the corresponding \( d \)-dimensional distribution by \( \phi^{(d)}(z_1, \ldots, z_d|R) \). \( R \) is a (positive-definite) correlation matrix of dimension \( d \times d \). We do not specify yet which correlation coefficient we use; this issue is reconsidered after the following formal elaborations.

By rearranging equation (2), we can express our copula density as follows (Clemen and Reilly, 1999):

\[
c(\Phi(z_1), \ldots, \Phi(z_d)|R) = \frac{\phi^{(d)}(z_1, \ldots, z_d|R)}{\phi(z_1) \times \cdots \times \phi(z_d)}
\]

(3)

In vector notation, \( z = (z_1, \ldots, z_d)^T \). Using the density expression for the (multivariate) normal distribution, this equation can be expressed as

\[
c(\Phi(z_1), \ldots, \Phi(z_d)|R) = \frac{\exp(z^T(R^{-1} - I)z/2)}{|R|^{1/2}}.
\]

(4)

Now we need to determine the joint density \( f(x_1, \ldots, x_d) \) for the observed (or simulated) univariate random variables \( X_1, \ldots, X_d \). Let us express the normal inverse as \( \Phi^{-1} \) and define \( Z_i = \Phi^{-1}(F_i(X_i)) \). Correspondingly,

\[
z = (\Phi^{-1}(F_1(x_1)), \ldots, \Phi^{-1}(F_d(x_d))).
\]

(5)

Note that this expression reflects a monotone transformation of the observed random variables. Using this definition we simply need to plug-in equation (4) into equation (2) which leads to

\[
f(x_1, \ldots, x_d|R) = f(x_1) \times \cdots \times f(x_n) \frac{\exp(z^T(R^{-1} - I)z/2)}{|R|^{1/2}}.
\]

(6)

This is how a normal copula is build-up. As we see, a normal copula is based on monotonic transformations of a multinormal distribution. This idea goes back to the old concept of “strained” multinormals introduced by Yule (1912). In fact, Gaussian copulas can be considered as “strained” multinormal distributions.

The question at this point is how these results could be used for non-normal data generation for SEM. We can easily simulate random variables \( X_1, \ldots, X_d \) independently from each other by drawing from known continuous univariate distributions with densities \( f(x_1), \ldots, f(x_d) \) and distribution function \( F(x_1), \ldots, F(x_d) \). Each random variable is transformed by the normal inverse according to the monotone transformation in equation (5). This is the first component in our simulation setting.

The second component captures the dependency between the random variables \( X_1, \ldots, X_d \). We need to take up this discussion in detail since SEM are usually computed on the basis of Pearson correlations. The Pearson product-moment correlation has two undesirable properties within this context: First, it depends on the margins, and second, it is not invariant under monotone transformations. Therefore, if we use the Pearson correlation, \( R \), on the left hand side of equation (6), is different from the one in our multivariate normal specification \( \phi^{(n)}(z_1, \ldots, z_d|R) \). A dependence measure that is invariant under monotonic transformations and does not depend on the marginal distribution is Spearman’s rank correlation coefficient \( \rho \). Alternatively, we could also use Kendall’s \( \tau \).
For this reason, within a Gaussian copula framework, the matrix $R$ typically consists of Spearman correlations. This creates somewhat of a problem for SEM simulation since, as mentioned above, the estimation is typically based on Pearson correlations (or simple covariances). There exist back and forth transformations between Spearman and Pearson correlations (Kruskal, 1958). These transformation work for the bivariate normal case only (see Schweizer and Wolff, 1981, for a thorough discussion) which is not relevant within our non-normal context, of course. As a consequence, a simple application of Gaussian copulas for data simulation does not work for our SEM purposes. In this paper we will develop a more sophisticated data generation approach that involves, among others, Gaussian copulas.

In order to give the reader an idea about possible density shapes of Gaussian copulas with arbitrary continuous margins we can look at the following simple example. Note that an easy-to-use implementation for computing and simulating a wide range of copulas is given in Yan (2007) by means of the R (R Development Core Team, 2011) package copula. First, we specify a Spearman correlation of $\rho = 0.8$ and compute the bivariate normal copula density (which has uniform margins).

In a second step we use copula to “couple” a beta margin for $X_1$ with shape parameters $\alpha = 10$ and $\beta = 2$ and a gamma margin for $X_2$ with shape parameter $k = 4$ and scale parameter $\theta = 2$. This gives us the joint density function $f(x_1, x_2 | r)$. Now we can easily compute a 3D-plot of the joint density function and a corresponding contour plot.

### 2.2 Other types of multivariate copulas

Many of the copulas proposed in the literature are only bivariate. Due to our general multivariate setting, we will not consider them in this article. Rather, we present some
multivariate copulas that are relevant for non-normal multivariate data generation, without this list being claimed to be exhaustive.

As mentioned above, the Gaussian copula belongs to the class of elliptical copulas. Another type of copula that is part of this class is the Student’s $t$-copula. Another copula class are Archimedean copulas. Corresponding comprehensive elaborations can be found in (Nelsen, 2006, Chapter 4). Such copulas can be written in the form

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

(7)

where $\psi$ is the generator function (continuous strictly decreasing) with its corresponding inverse $\psi^{-1}$.

Archimedean copulas are widely used in applications due to their simple form, a variety of dependence structures, and other “nice” properties (Nelsen, 2005). Examples of Archimedean copulas are given in Table 1. Note that these copulas are one-parameter copulas with dependency parameter $\theta$.

<table>
<thead>
<tr>
<th>Family</th>
<th>Parameter Space</th>
<th>Generator $\psi(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clayton (1978)</td>
<td>$\theta \geq 0$</td>
<td>$t^{-\theta} - 1$</td>
</tr>
<tr>
<td>Frank (1979)</td>
<td>$\theta \geq 0$</td>
<td>$- \ln \frac{\exp(-\theta t) - 1}{\exp(-\theta - 1)}$</td>
</tr>
<tr>
<td>Gumbel (1960)</td>
<td>$\theta \geq 1$</td>
<td>$(- \ln t)^\theta$</td>
</tr>
</tbody>
</table>

The parameter $\theta$ captures the dependence between the margins and can be expressed by means of Kendall’s $\tau$. To illustrate possible shapes of Archimedean copulas we define two bivariate Frank copulas with parameters $\theta = 1$ and $\theta = 5$, respectively. As margins we pick a standard normal and an exponential distribution with $\lambda = 3$. The corresponding contour plots given in Figure 2.

A further type of multivariate copula that does not belong to the classes above is the Farlie-Gumbel-Morgenstern copula (FGM) which was discussed in Farlie (1960), Gumbel (1958), and Morgenstern (1956). The multivariate expression for the FGM-copula is

$$C(u_1, \ldots, u_d) = \prod_{i=1}^{d} u_i \left(1 + \sum_{1 \leq i < j} \theta_{ij}(1-u_i)(1-u_j)\right)$$

(8)

with $-1 \leq \theta_{ij} \leq 1$. The parameter approximates Spearman and Pearson correlations for the bivariate margins (Mari and Kotz, 2001). Here we present two bivariate FGM copulas with a $t$-distribution ($df = 5$) and a Cauchy distribution (location parameter $x_0 = 1$, scale parameter $\gamma = 0.5$).

The contour plots in Figure 3 shows the effect of two different FGM dependency parameterizations (margins are the same). The left hand side shows a positive dependence structure ($\theta = 0.9$) whereas the right hand side a negative one ($\theta = -0.9$). They are produced by the following lines:

3 A two-stage data generating process

3.1 Theory

As mentioned, for SEM we need to pose a true VC-structure on the data if we want to perform simulation studies. As we have seen, the Gaussian copula can only cope
Frank ($\theta=1$)

Frank ($\theta=5$)

FGM ($\theta=0.9$)

FGM ($\theta=-0.9$)

Figure 2: Frank copula with normal-exponential margins.

Figure 3: FGM copula with t-Cauchy margins.
with Spearman correlations. The other types of multivariate copulas do not have a covariance matrix as parameter, rather they have dependency parameters that can be expressed in terms of Spearman’s $\rho$ or Kendall’s $\tau$ (Joe, 1997; Mari and Kotz, 2001). Thus, we need the solve this issue by means of the following considerations.

Let us first look at the following formal relations in SEM. Assume that $X$ is our (simulated) dataset. The corresponding empirical product-moment covariance matrix is $S = \text{cov}(X)$. Now, we specify a certain model covariance matrix $\Sigma_0$. It can be either equal to $S$, or it can be specified according to some hypothesis (e.g., based on SEM specification $\Sigma_0 = \Lambda \Lambda' + \Psi$), or it can have whatever values of interest as long as it is positive definite.

Having now our matrices $X$, $S$, and $\Sigma_0$, we apply the following transformation:

$$Y = XS^{-\frac{1}{2}}\Sigma_0^\frac{1}{2}. \quad (9)$$

Note that this reflects a linear transformation such that the covariance of $Y$ results in

$$Y'Y = \Sigma_0^\frac{1}{2}S^{-\frac{1}{2}}X'XS^{-\frac{1}{2}}\Sigma_0^\frac{1}{2} = \Sigma_0^\frac{1}{2}SS^{-\frac{1}{2}}\Sigma_0^\frac{1}{2} = \Sigma_0^\frac{1}{2}I\Sigma_0^\frac{1}{2} = \Sigma_0. \quad (10)$$

The problem with this simple transformation is that the rows in $Y$ are not independent and identically distributed ($iid$) random variables.

In order to achieve the $iid$ property of the rows of $Y$, we propose the following two-stage approach: First, we have a warm-up stage where we simulate the data matrix $X_w$ (warm-up sample) of dimension $n_w \times d$ according to a multivariate copula from the previous section and compute the covariance matrix $S_w$. The corresponding sample size $n_w$ needs to be large since we want to approximate the true copula covariance structure. Note that if we have formal expressions for the true covariance structure of the corresponding copula, we do not need to perform the warm-up stage.

Second, we propose a production stage where we produce an additional sample which is subject to the transformation above. Let $\Sigma_0$ be the model covariance matrix. Now, we draw another sample from the same copula specification which we call production sample $X_p$. The sample size $n_p$ does not necessarily need to be large; it reflects the sample size of our final SEM input data. Subsequently, we apply equation (9) according to

$$Y = X_pS_w^{-\frac{1}{2}}\Sigma_0^\frac{1}{2}, \quad (11)$$

where we use the sample $X_p$ from the production stage and the covariance matrix $S_w$ from the warm-up stage. The rows of $X_p$ are clearly $iid$ because it was drawn from the copula independently from the remaining covariance specifications in this equation. Furthermore, since equation (11) is a linear transformation of $X_p$, the corresponding random variables in $Y$ are $iid$ as well. As a result, the data matrix $Y$ reflects multivariate non-normal data that correspond to the VC-structure in $\Sigma_0$.

A striking advantage of this approach is that we can use whatever multivariate copula we want for data generation; we are not limited to Gaussian copulas at all. The degree of multinormal violation depends on the copula specifications (type of copula, type of margins). Since there is basically no limitations in the use of copulas, we can also create non-elliptical multivariate distributions. Using Monte Carlo techniques, the next section explores whether the VC-structure of our simulated $Y$ dataset really reflects the model VC-structure $\Sigma_0$. 

9
3.2 Assessing the accuracy of the data generation approach

In this section we explore how much the parameter $n_w$ should be tuned to guarantee a sample comes from a population with the desired pre-specified covariance matrix. For that we will simulate considerably large samples. This does not imply that for a given application the sample size has to be large. Here, the large samples are used only for purpose of accuracy of the assessment of the quality of our data generating procedure, the accuracy in reproducing a population with an specified covariance matrix. In assessing this accuracy, we will vary the tuning parameter $n_w$, the sample size $n_p$, and also the class of copula distributions. We will show that the variation of the marginal distributions of the copula and their dependency structure will affect the quality of the accuracy.

According to the theory above, the VC-structure of $Y$ denoted by $S_Y$ (and $\Sigma_Y$, respectively, as the population equivalent) should reflect the model covariance structure $\Sigma_0$. Therefore, we need to test:

$$H_0: \Sigma_Y = \Sigma_0.$$ 

Note that this hypothesis implies a one-sample test on the VC-matrix $\Sigma_Y$. Since we are creating non-normal data we need to make sure that the corresponding test is robust against normality violations. Testing this hypothesis under these assumptions is not trivial. But we can achieve a reliable testing procedure within an SEM context, as will be shown.

An example for an EQS (Bentler, 2006) syntax file that performs such a test is given in Appendix A1. For a 3-dimensional data set ($n_p = 12000$), we define a 3-factor model where each indicator loads to one factor only. We fix the variances of the factors and the covariances between the factors. Obviously, there are no parameters to estimate; we just want to extract the $\chi^2$-value based on the asymptotical distribution-free (ADF, Browne (1984); in EQS denoted as AGLS) approach after one iteration. Note that for large sample sizes, ADF theory states that the ordinary $\chi^2$-statistic is unbiased for whatever normality violations. Since we claim that we create data that obey $\Sigma_0$, the model should fit and thus the $\chi^2$-test should not reject the null hypothesis above.

In order to study this topic in a comprehensive manner, we establish the following simulation setting. First, we need to determine which values and specifications might influence the performance of the $\Sigma_0$ approximation by $S_Y$. Second, we will look at at various types of copulas. As a benchmark we include a normal copula (NC) with normal margins (i.e., a multivariate normal distribution). In addition, we include Gumbel copulas (GC) and Clayton copulas (CC).

Second, for GC and CC, we vary the dependency structure of the margins through $\theta$: For the GC we choose marginal independence through $\theta = 1$ (that corresponds to Kendall’s $\tau = 0$; according to $\tau = (\theta - 1)/\theta$, medium dependence through $\theta = 2$ ($\tau = 0.5$), and strong dependence through $\theta = 10$ ($\tau = 0.9$). For the CC we choose marginal independence through $\theta = 0$ ($\tau = 0$ according to $\tau = \theta/(2 + \theta)$), low-medium dependence $\theta = 0.5$ ($\tau = 0.2$), and medium-high dependence $\theta = 5$ ($\tau = 0.71$).

Third, we vary the complexity of the copulas in terms of various margin specifications. Pertaining to the margins, the first scenario uses normal margins that suggest a low violation from multivariate normality ($\mu_1 = 0$, $\mu_2 = 1$, $\mu_3 = 3$; $\sigma_1 = \sigma_2 = \sigma_3 = 1$). The second scenario uses exponential margins which clearly violate normality. But still, the one-parametric exponential distribution is still easy to handle ($\lambda_1 = 0.5$, $\lambda_2 = 1$, $\lambda_3 = 2$). For the third scenario we choose rather complex margins: The first margin is
Weibull distributed (shape $k = 0.5$, scale $\lambda = 1$), the second margin Beta distributed ($\alpha = 1$, $\beta = 0.5$), and the third margin is Gamma distributed (shape $k = 4$, scale $\theta = 0.5$).

Considering the 2-step approach above, the parameters of interest are the sample sizes $n_w$ and $n_p$. For the warm-up sample, we claim that $n_w$ needs to be large since we want to approximate the true VC-structure of the copula. Therefore we start with a moderate value of $n_w = 1e5$, increase it to $n_w = 1e6$, and, finally, consider a very large warm-up sample size of $n_w = 1e7$. Since we are using ADF, the production sample $n_p$ which reflects the final sample size of the data, should be somewhat large as well. According to ADF theory, it has been claimed in the literature that the sample size needs to be larger than 5000 (see, e.g., Hu et al., 1992). Thus, in one simulation series we use a low value of $n_p = 1000$ and increase this value in the next two series to $n_p = 6000$ and $n_p = 12000$, respectively.

Eventually, we get 144 simulation scenarios (we are not doing CC with normal margins). For each scenario 1000 simulation runs (i.e., data replications) are performed. At the end we compute the rejection rate for each particular scenario. We can state that the data generating process works properly (i.e., the generated data reflect the true $\Sigma_0$ structure), if an $\alpha$-level of 0.05 is held. The model VC-matrix $\Sigma_0$ is specified as follows:

$$\Sigma_0 = \begin{pmatrix}
1 & 0.8 & 0.2 \\
0.8 & 1 & 0.5 \\
0.2 & 0.5 & 1
\end{pmatrix}.$$

The core of the analysis is performed in R (R Development Core Team, 2011). For all copula specifications and the corresponding random number generation we use the \texttt{copula} package (Yan, 2007). For the two-stage (warm-up/production) approach as described in Section 3.1 we wrote our own R code. It is given in Appendix A2. For the determination of the $\chi^2$-value for testing the VC-equality hypothesis we use EQS (Bentler, 2006). The corresponding syntax file given in Appendix A1. Note that we make use of the \texttt{REQS} package (Mair et al., 2010) that allows to perform an SEM estimation call from within the R environment. After the estimation, the package imports the EQS results back into R. As a final step we extract the $p$-value of the $\chi^2$-test and count the number of rejections. This leads to the rejection rates given in Table 2.

In this table, the first number in brackets pertaining to the name of the copula, denotes the $\theta$ parameter; except for the NC where we choose

$$\Sigma^* = \begin{pmatrix}
1 & 0.9 & 0.5 \\
0.9 & 1 & 0.3 \\
0.5 & 0.3 & 1
\end{pmatrix}.$$ 

The subsequent bracket expression denotes the margin specification: normal margins (N-N-N), exponential margins (E-E-E), and Weibull-Beta-Gamma margins (W-B-G).

Now let us examine the rejection rates in Table 2 a bit closer. We see that for the NC the rates are pretty much constant across the scenarios. Hence, for this multivariate normal case (specified through copulas), no large sample sizes are needed. For the GC with normal margins the situation is similar. We get satisfactory rejection rates already for low $n_p$ and $n_w$, they depend slightly on the amount of dependency specified through $\theta$. For the GC with exponential margins we observe that the rejection rates for $n_p = 1000$ are consistently larger than 0.10. Increasing $n_p$ leads to satisfactory rejection rates again. For the GC with W-B-G margins the rejection rates are critical throughout the simulation settings; especially for $\theta = 10$. 

11
Table 2: Rejection rates for the two-stage data generating process (1000 runs each)

<table>
<thead>
<tr>
<th>nw: 1e5</th>
<th>1e6</th>
<th>1e7</th>
</tr>
</thead>
<tbody>
<tr>
<td>np: 1000</td>
<td>0.060</td>
<td>0.064</td>
</tr>
<tr>
<td>6000</td>
<td>0.064</td>
<td>0.071</td>
</tr>
<tr>
<td>12000</td>
<td>0.080</td>
<td>0.065</td>
</tr>
<tr>
<td>GC(10; N-N-N)</td>
<td>0.098</td>
<td>0.072</td>
</tr>
<tr>
<td>1e6</td>
<td>0.064</td>
<td>0.071</td>
</tr>
<tr>
<td>1e7</td>
<td>0.080</td>
<td>0.065</td>
</tr>
<tr>
<td>GC(10; N-N-N)</td>
<td>0.098</td>
<td>0.072</td>
</tr>
<tr>
<td>GC(1); N-N-N</td>
<td>0.076</td>
<td>0.071</td>
</tr>
<tr>
<td>GC(2); N-N-N</td>
<td>0.063</td>
<td>0.065</td>
</tr>
<tr>
<td>GC(10); N-N-N</td>
<td>0.098</td>
<td>0.072</td>
</tr>
<tr>
<td>GC(1); E-E-E</td>
<td>0.122</td>
<td>0.066</td>
</tr>
<tr>
<td>GC(2); E-E-E</td>
<td>0.126</td>
<td>0.070</td>
</tr>
<tr>
<td>GC(10); E-E-E</td>
<td>0.111</td>
<td>0.072</td>
</tr>
<tr>
<td>GC(1); W-B-G</td>
<td>0.191</td>
<td>0.119</td>
</tr>
<tr>
<td>GC(2); W-B-G</td>
<td>0.252</td>
<td>0.147</td>
</tr>
<tr>
<td>GC(10); W-B-G</td>
<td>0.432</td>
<td>0.250</td>
</tr>
<tr>
<td>CC(0); E-E-E</td>
<td>0.136</td>
<td>0.078</td>
</tr>
<tr>
<td>CC(0.5); E-E-E</td>
<td>0.108</td>
<td>0.069</td>
</tr>
<tr>
<td>CC(5); E-E-E</td>
<td>0.171</td>
<td>0.088</td>
</tr>
<tr>
<td>CC(0); W-B-G</td>
<td>0.198</td>
<td>0.114</td>
</tr>
<tr>
<td>CC(0.5); W-B-G</td>
<td>0.210</td>
<td>0.122</td>
</tr>
<tr>
<td>CC(5); W-B-G</td>
<td>0.232</td>
<td>0.127</td>
</tr>
</tbody>
</table>

Similar to GC, the CC simulations with exponential margins lead to acceptable rejection rates as \( n_p \) and \( n_w \) increase. Note that for the CC with W-B-G margins we use lower dependency parameter as for GC, i.e., no, low-medium, and medium-high dependency. Correspondingly, the rejection rates are closer to \( \alpha = 0.05 \) compared to the GC setting. However, \( n_p \) and \( n_w \) need to be large such that the test keeps (almost) the \( \alpha \)-level.

To summarize, in this section we have shown, computationally, that our approach works and that the rejection rates for our one-sample test on the VC-matrices are close to \( \alpha = 0.05 \). We will take up again the case for complex marginal specifications with possibly highly correlated margins in the Discussion.

4 Example: A two-factor model

As a simple example that could reflect a prototype SEM simulation scenario, we generate multivariate non-normal data involving 6 indicators. We specify a 2-factor model where indicator 1 to 3 load to the first factor and indicator 4 to 6 load to the second. The corresponding model covariance matrix is specified as follows:

\[
\Sigma_0 = \begin{pmatrix}
1.00 & 0.80 & 0.80 & 0.20 & 0.20 & 0.20 \\
0.80 & 1.00 & 0.80 & 0.20 & 0.20 & 0.20 \\
0.80 & 0.80 & 1.00 & 0.20 & 0.20 & 0.20 \\
0.20 & 0.20 & 0.20 & 1.00 & 0.80 & 0.80 \\
0.20 & 0.20 & 0.20 & 0.80 & 1.00 & 0.80 \\
0.20 & 0.20 & 0.20 & 0.80 & 0.80 & 1.00 \\
\end{pmatrix}
\]

The copula setting we use is a 6-dimensional Clayton copula with \( \theta = 3 \). As margins, we choose a W-B-G-W-B-G specification with the following parameterization: Weibull
Figure 4: Distribution approximation for ordinary $\chi^2$- and Satorra-Bentler $\chi^2$-statistic.

(\text{shape } k = 0.5, \text{scale } \lambda = 1), \text{Beta (}\alpha = 1, \beta = 0.5\text{), Gamma (}\text{shape } k = 1, \text{scale } \theta = 3\text{), Weibull (}\text{shape } k = 0.5, \text{scale } \lambda = 1\text{), Beta (}\alpha = 1, \beta = 0.5\text{), and Gamma (}\text{shape } k = 1, \text{scale } \theta = 3\text{). As we see, both sets of indicators have the same marginal specification.}

For these copula and Sigma settings we perform 100 sample replications ($n_p = 1000, n_w = 100000$), fit the SEM using robust ML as estimation method according to the \textit{EQS} syntax file given in Appendix B1, and extract the ordinary and Satorra-Bentler $\chi^2$-statistic (Satorra and Bentler, 1994). Finally, we produce Q-Q-plots that plot the theoretical $\chi^2$ quantiles ($df = 8$) against the quantiles from the test statistics. We expect that the distribution of the Satorra-Bentler $\chi^2$ is more robust against this normality violation than the approximation of the ordinary $\chi^2$ statistic. The full \texttt{R} code, including the \texttt{REQS} call, is given in Appendix B2. Note that the \texttt{CopSEM()} function, given in Appendix A2, is required to be sourced first into the \texttt{R} environment.

The main output is the two Q-Q-plots given in Figure 4. We see clearly that the $\chi^2$-approximation for the ordinary $\chi^2$-test is not satisfactory. There is a clear bias due to normality violations in the data. The Satorra-Bentler $\chi^2$, however, approximates the $\chi^2$-distribution fairly well and, hence, this result suggests once more that this statistic is robust to normality violations, even if the margins are specified by copulas. Of course, a thorough examination of this result using different types of copulas would be needed in order to make a clear conclusion. This exceeds the scope of the paper, however.
5 Discussion

This paper presents a copula based two-stage data generating approach for simulating non-normal multivariate data that correspond to a certain underlying VC-structure. These data can be used as input data for SEM. Taking into account the amount of different copulas that have been proposed in the literature over the last 50 years, the flexibility of this approach becomes obvious.

The test used to assess the accuracy of the procedure assumes a multivariate normal distribution for the vectorized form of the sample covariance matrix. In the case of heavy tailed distributions as the one contemplated by some copulas, this may only hold for tremendously large sample sizes. This may explain the rejection rates above the \( \alpha \)-level that is seen in Table 2 for some of the copulas, specially, when the marginals have not the same distribution. Finiteness of moments up to order four is also assumed in the ADF test. Statistical theory for finiteness of moments in distributions obtained by copula is an issue that needs to be explored further.

Table 2 that informs on the accuracy of the procedure proposed has been demanding in CPU computer time, since Monte Carlo analysis involving replications of an ADF test in SEM for a non-normal population with variation of the population (classes of copulas), the “warm-up” parameter \( n_w \), and sample size \( n_p \), has been contemplated. In the use of the method in routine Monte Carlo evaluation of SEM methods, the matrix \( S_w \) needs to be computed only once for all the Monte Carlo replications (as the population is kept constant). Thus, our method would not increase the usual requirements on CPU of typical Monte Carlo evaluations in SEM.

References


normality on the estimation of mediated effects in latent variable models. *Structural


London, UK.


structural equation modeling. *Structural Equation Modeling: A Multidisciplinary


London.


Morgenstern, D. (1956). Einfache Beispiele zweidimensionaler Verteilungen [Simple ex-
amples of two-dimensional distributions]. *Mitteilungsblatt für Mathematische Statis-

report, Department of Mathematical Sciences, Lewis & Clark College.


R Development Core Team (2011). *R: A Language and Environment for Statistical
Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-
900051-07-0.

for simulation of structural equation models using Mattson’s method. *Multivariate
Behavioral Research*, 37:227–244.


Appendix A1: EQS syntax file for VC-test

The following EQS syntax reflects an example for testing the equality of covariance matrices (1-sample test) for non-normal data within an SEM framework.

```
/TITLE
Test Covariance Matrices
/SPECIFICATIONS
CAS=12000; VAR=3; ME=AGLS; data='covtest.dat';
matrix = raw;
/EQUATIONS
V1 = F1;
V2 = F2;
V3 = F3;
/VAR
F1 = 1; F2 = 1; F3 = 1;
/COV
F1,F2 = .8;
F2,F3 = .5;
F1,F3 = .2;
/OUTPUT
parameter;
standard error;
codebook;
listing;
data = 'covtest.ets';
/END
```
Appendix A2: Two-stage data generating process in R

The following function implements the two-stage data generating process as described in Section 3.1.

```r
CopSEM <- function(copmvdc, Sigma, nw = 100000, np = 1000) {
  ## copmvdc ... joint density from mvdc()
  ## Sigma ... model VC-matrix to be approximated
  ## nw ... sample size for warm-up sample
  ## np ... sample size for production sample

  Xw <- rmvdc(copmvdc, nw) ## draw warm-up sample
  Sw <- cov(Xw) ## warm-up VC matrix

  Sigma.eigen <- eigen(Sigma) ## EV decomposition Sigma
  Sigmaroot <- Sigma.eigen$vectors %*% sqrt(diag(Sigma.eigen$values))
  + %*% t(Sigma.eigen$vectors) ## root Sigma

  Sx.eigen <- eigen(solve(Sw)) ## EV decomposition S
  Sxroot <- Sx.eigen$vectors %*% sqrt(diag(Sx.eigen$values))
  + %*% t(Sx.eigen$vectors) ## root S

  X <- rmvdc(copmvdc, np) ## draw production sample
  Y <- (X %*% (Sxroot) %*% Sigmaroot) ## linear combination for Y

  list(Y = Y, covY = (cov(Y))) ## return Y and cov(Y)
}
```

The function returns the generated data matrix $Y$ of dimension $n_p \times d$ where $d$ is the dimension of the data specified through the dimension of $\Sigma$. The argument `copmvdc` is an object created by the `mvdc()` function from the `copula` package. Here is a simple example:

```r
## Sigma specification
Sigma <- matrix(diag(rep(1,3)), ncol = 3)
Sigma[lower.tri(Sigma)] <- c(0.8, 0.2, 0.5)
Sigma <- t(Sigma)
Sigma[lower.tri(Sigma)] <- c(0.8, 0.2, 0.5)

## Copula specification (3D Gumbel, theta = 2, exponential margins)
coppar <- gumbelCopula(2, dim = 3)
copjoint <- mvdc(coppar, margins = c("exp","exp","exp"),
                + paramMargins = list(list(rate = 0.5),
                + list(rate = 1), list(rate = 2)))

## Apply two-stage function
res.sim <- CopSEM(copjoint, Sigma, nruns = 1000, np = np, nw = nw)
data <- res.sim$Y
data.cov <- res.sim$covY
```
The object data now contains non-normal data that obey the VC-structure defined in Sigma.
Appendix B1: EQS syntax for 2-factor example

Below is the EQS syntax file for the 2-factor example in Section 4.

/TITLE
2-factor model
/SPECIFICATIONS
CAS=1000; VAR=6; ME=ML,Robust; data='2factorreqs.dat';
matrix = raw;
/EQUATIONS
V1 = *F1+ E1;
V2 = *F1+ E2;
V3 = *F1+ E3;
V4 = *F2+ E4;
V5 = *F2+ E5;
V6 = *F2+ E6;
/VARIANCES
F1 = 1; F2 = 1; E1 TO E6 = *;
/COVARIANCES
F1, F2 = *;
/OUTPUT
parameter;
standard error;
codebook;
listing;
data = '2factorreqs.ets';
/END
Appendix B1: R code for 2-factor example

Note that the CopSEM() function from Appendix A2 need to be sourced first into R

```r
require("copula")
require("REQS")

## copula specification
NC.sem <- claytonCopula(3, dim = 6)
rccop.sem <- mvdc(NC.sem, margins = c("weibull","beta","gamma",
+ "weibull","beta","gamma"),
+ paramMargins = list(list(shape = 0.5, scale = 1),
+ list(shape1 = 1, shape2 = 0.5),
+ list(shape = 1, rate = 0.5, scale = 3),
+ list(shape = 0.5, scale = 1), list(shape1 = 1, shape2 = 0.5),
+ list(shape = 1, rate = 0.5, scale = 3)))

## Sigma specification
vc.struc <- c(0.8, 0.8, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
+ 0.2, 0.8, 0.8, 0.8)
Sigma <- matrix(diag(rep(1,6)), ncol = 6)
Sigma[lower.tri(Sigma)] <- vc.struc
Sigma <- t(Sigma)
Sigma[lower.tri(Sigma)] <- vc.struc

nrep <- 100               ## number of replications

sbvec <- NULL
chisqvec <- NULL
for (i in 1:nrep) {
  Y <- CopSEM(rccop.sem, Sigma)$Y  ## data from 2-stage approach
  ## now run the procedure
  eqs.fit12 <- run.eqs(EQSpgm = "C:/Program Files (x86)/eqs62/wineqs.exe",
    EQSmodel = "2factorreqs.eqs", serial = "1234",
    Rmatrix = Y, datname = "2factorreqs.dat")
  sb.chisq <- eqs.fit12$fit.indices["SBCHI",]
  chisq <- eqs.fit12$fit.indices["CHI",]
  df.F12 <- eqs.fit12$model.info[5,]
  sbvec <- c(sb.chisq, sbvec)
  chisqvec <- c(chisq, chisqvec)
}

chisq.data <- rchisq(1000, df = df.F12)  ## generate chi^2-data

## Q-Q-plots
ax.limits <- range(c(sbvec, chisqvec, chisq.data))
par(mfrow = c(1,2))
```

qqplot(chisqvec, chisq.data, xlab = "observed quantiles", ylab = "theoretical quantiles")
qqplot(sbvec, chisq.data, xlab = "observed quantiles", ylab = "theoretical quantiles")
abline(0,1)